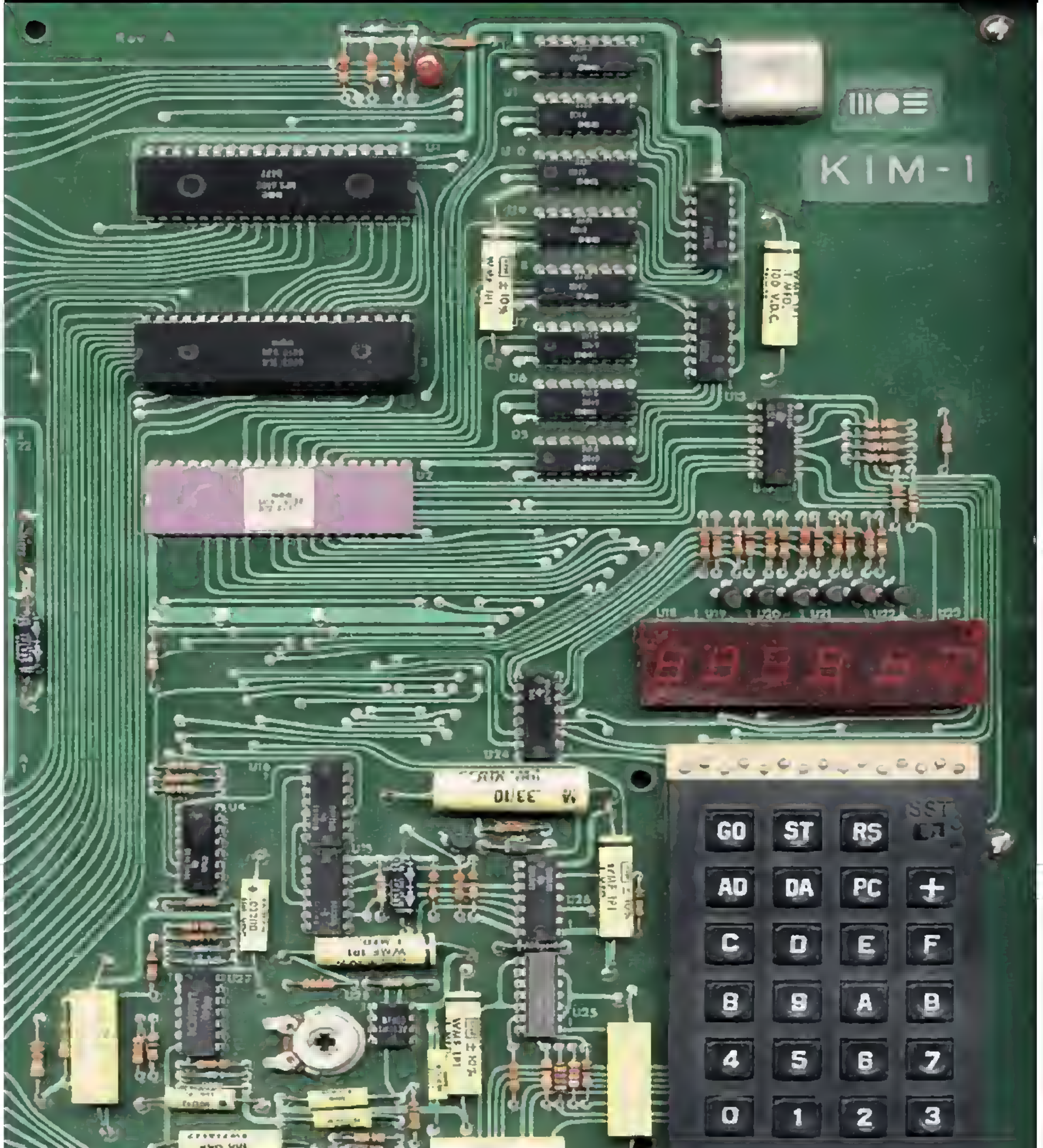
Jurassic News

Retrocomputer Magazine

Anno 4 - Numero 26 - Novembre 2009



Jurassic News

Rivista aperiodica di Retro-computing

Coordinatore editoriale
Salvatore Macomer [Sm]

Redazione
Sonicher [Sn]
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Tullio Nicolussi [Tn]
Lorenzo 2 [L2]
Besdelsec [Bs]
Lorenzo Paolini [Lp]

Impaginazione e grafica
Anna [An]

Diffusione
marketing@jurassicnews.com

La rivista viene diffusa in formato PDF via Internet agli utenti registrati sul sito www.jurassicnews.com. la registrazione è gratuita e anonima; si gradisce comunque una registrazione nominativa.

Contatti
info@jurassicnews.com

Jurassic News promuove la libera circolazione delle idee

Novembre 2009

In Copertina

La piastra KIM-1 della MOS Technology. Una delle prime piastre a microprocessore adatte anche ad un uso hobbystico.

Editoriale

Riviste di elettronica e Internet

Avrete notato un cambiamento nella copertina della nostra rivista: mancano "gli strilli", come vengono chiamati i titoli che hanno lo scopo di richiamare l'attenzione del potenziale lettore al contenuto della rivista.
Abbiamo deciso di provare ad eliminarli. In fondo non abbiamo bisogno di catturare l'attenzione perché chi ama Jurassic News lo scarica e lo consulta anche se il contenuto gli è in qualche modo sconosciuto.
Poi c'è la questione del Web; sì, perché come nel nostro caso è facile scoprire i contenuti di un fascicolo: basta andare nella sezione "Indice" del sito per avere il dettaglio del contenuto di ogni fascicolo pubblicato.
La fruizione di una rivista su Web pian pianino sta cambiando le regole del gioco e questo è vero anche per le riviste "tradizionali", cioè su carta e distribuite in edicola.
Sono sempre più convinto che non c'è futuro per l'editoria specializzata (almeno per quella elettronica) nella distribuzione in edicola. Veramente ha poco senso riempirsi la casa di carta e parimenti è costoso per l'editore procedere nei canali tradizionali di distribuzione.
Certo c'è tutta la questione "furto", cioè il fatto che se una rivista è elettronica, diventa fin troppo facile farne delle copie per gli amici...
Recentemente si è ri-affacciata la linea editoriale della Infomedia (http://www.infomedia.it).
Per noi che abbiamo vissuto in prima persona la programmazione del PC negli anni '90, l'Infomedia è stata un riferimento. Poi improvvisamente l'oblio, inevitabile, perché chi può permettersi oggi di aspettare che te le spieghino le nuove tecnologie: vai sul Web e ti leggi tutto di tutto. Questa è la strada delle riviste specializzate, inutile puntare i piedi.

[Sm]

Jurassic News

è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Retrocomputing

Il progetto ReFUN

Dove si parla di un progetto di "Recupero Finzionale" di vecchi calcolatori.

P.L.U.T.O.

Free Software Project since 1992

PLUTO | Journal | ILDP | ReFUN | Scuola | Meeting | Web | Login | Cerca

Home

PLUTO ReFUN

Coordinatore: Alberto Cammozzo, mmzz(at)pluto.it
Vice coordinatore: Danilo Selvestrel, danny(at)pluto.it
Mailing list: pluto-refun

Il progetto ReFUN si occupa del recupero funzionale di vecchi calcolatori ripristinando l'hardware e installando software libero.

Dividiamo i computer da recuperare in due categorie distinte:

1. macchine di qualità o di un qualche interesse per la storia dell'informatica come workstation o server Digital, Sun, HP Apollo, solitamente molto robuste ed affidabili. Possono essere impiegate come server di fascia bassa o come componenti di *cluster*.
2. vecchi computer di classe "PC" (286, 386, 486, ...) che possono ancora essere utilizzati per attività non di punta (composizione testi, lettura email, apparati di rete), con sistemi operativi liberi 'alleggeriti'.

Oltre che re-cupero fun-zionale **ReFUN** si puo' leggere anche così: **Re** sta per *"di nuovo"* e anche per *"retrocomputing"*, e **FUN** sta per divertimento, indicando che quest'attività di retrocomputing serve a far vivere di nuovo i vecchi computer e viene svolta per divertimento.

1 | 2 | 3 | 4 | 5 | seguente | tutte

PLUTO Project

- Il PLUTO
- Progetti
- Contatti
- Mailing list
- Mirror
- Sponsor
- Notizia
- CVS

Progetti PLUTO

- Journal
- ILDP
- ReFUN
  - Criteri
  - Come Lavoriamo
  - Laboratori
  - Macchine
- Scuola
- Meeting
- Web

Il recupero delle vecchie macchine di calcolo è ovviamente l'anima stessa dell'hobby chiamato Retro Computing, o meglio ne rappresenta la parte "hardware" che si completa poi con l'uso delle macchine recuperate, normalmente utilizzando il software originale.

Come in ogni hobby che si rispetti anche nel retro computing ci sono i "puristi", cioè coloro che rispettano alla lettera sia l'hardware che il software dei sistemi recuperati. C'è però chi la pensa diversamente e che persegue l'idea del cosidetto "recupero funzionale".

Azzardando una definizione operativa si potrebbe riassumere l'attività con: -"Il Recupero Funzionale" consiste nel rendere funzionante un sistema di calcolo obsoleto sostituendone eventualmente alcune parti hardware con equivalenti o migliorative e utilizzando software il più possibile funzionale alle esigenze moderne".

Abbiamo incontrato un vero esperto di questa pratica il quale ha fondato un gruppo che si dedica a questa attività. Si tratta di Alberto Cammozzo, responsabile e cordinatore del progetto ReFUN e da lui stesso sentiamo in cosa consiste e come si realizza la loro idea.

- Il progetto ReFUN si occupa del recupero funzionale di vecchi calcolatori ripristinando l'hardware e installando software libero.

Dividiamo i computer da recuperare in due categorie distinte:

1. macchine di qualità o di un qualche interesse per la storia dell'informatica come workstation o server Digital, Sun, HP Apollo, solitamente molto robuste ed affidabili. Possono essere impiegate come server di fascia bassa o come componenti di cluster.

2. vecchi computer di classe "PC" (286, 386, 486, ...) che possono ancora essere utilizzati per attività non di punta (composizione testi, lettura email, apparati di rete), con sistemi operativi liberi 'alleggeriti'.

Oltre che re-cupero fun-zionale ReFUN si puo' leggere anche così: Re sta per "di nuovo" e anche per "retrocomputing", e FUN sta per divertimento, indicando che quest'attività di retrocomputing serve a far vivere di nuovo i vecchi computer e viene svolta per divertimento.

C'era una volta...

Quasi quindici anni fa scoprii una miniera nella cantina del posto dove lavoravo: perforatrici di schede, terminali, stampanti, un vecchio Apple II (oggetto del mio desiderio negli anni Ottanta), cavi gialli ethernet coi relativi "vampiri", tutto accatastato e coperto di polvere, in attesa di essere scaricato dall'inventario e affidato alle rudi cure di Italmaceri. Infatti poco tempo dopo ebbi il dispiacere di veder buttare con malagrazia nei camion quei rottami pagati fortune pochi anni prima. Non che fossi contrario per principio a buttare vecchie macchine, ma ho sempre avuto molta stima per l'ingegno umano e per i suoi frutti: gli uomini passano, ma le opere delle loro mani e delle loro menti restano. Quelle macchine rappresentavano chi le aveva fatte, e alcune di esse erano fatte proprio bene.

In particolare ero incantato dalla straordinaria cura nella progettazione e realizzazione di apparecchi fatti per resistere, durare, essere efficienti. Soluzioni che non vedevo nelle macchine sulle quali allora lavoravo, nemmeno su quelle più costose. Un esempio: insieme ad altri colleghi "archeologi" disseppelliamo dalla cantina un DEC Professional, lo apriamo e scopriamo che le schede si adattano alla loro basetta con dei contatti a ganascia che vengono serrati da una leva girevole: che differenza rispetto alle schede ISA da inserire a forza nella motherboard scricchiolante! La CPU, montata su una scheda molto pulita, era composta da due chip separati. Strano. Accendiamo la macchina, silenziosissima, e questa fa il solito test delle ram e segnala un guasto, mostrando un esploso in grafica 3D della macchina con evi-

Jurassic Programmers è stato pubblicato nel numero di gennaio 2008 del periodico Redmond Developer News, nella colonna DevDisasters curata da Alex Papadimoulis.

denziata la scheda col chip difettoso. Mai più visto nulla di simile! E a quel tempo litigavamo con dei 386 assemblati particolarmente mal riusciti.

Il MUSI

Qualche anno dopo è stata una vera sorpresa scoprire che a Padova esiste un museo dell'informatica curato da Francesco Piva, un eclettico ed infaticabile amante della scienza, della natura e dell'uomo che raccoglie vecchi calcolatori, software e manuali sotto il tetto del museo didattico di storia dell'informatica (MUSI presso FWT-UNESCO) in una straordinaria collezione, nonostante gli ambienti poco adatti alle macchine e a chi desidera visitarle (attualmente la visita è di fatto impossibile per inagibilità dei locali). Ma questa è un'altra storia, quella della CLAC di Padova, piena di altri "nonostante" e di battaglie per scavalcarli e sopravvivere nonostante i "nonostante".

Nello stesso straordinario, apparentemente inospitale ma umanamente accoglientissimo ambiente si trova il laboratorio padovano del PLUTO, che già conoscevo dalla sua fondazione, ma al quale non avevo mai aderito (per i colori assurdi della pagina web dalla quale avrei dovuto iscrivermi). Danilo, un amico e collega dissotterratore di tesori mi ha invitato nel 2002 a esplorare gli antri del museo pieni di preziosi rottami: abbiamo così iniziato trascinando nella sede PLUTO Padova (detta "cuccia") un Digital, più precisamente un microVaX, se non ricordo male, e vi abbiamo installato sopra OpenBSD. Questa esperienza è stata e resta per noi diversa dalle altre quotidiane installazioni: nessun floppy disk, niente CD: bisogna fare tutto via rete, studiando i protocolli (bootp, rarp, MOP, tftp, ecc...). E solo dopo aver pulito bene l'hardware ed essersi documentati approfonditamente su vizi e virtù della macchina e scoperto cosa è supportato e cosa non lo è: a volte ci sono i driver per il frame-buffer, a volte nemmeno quelli per il disco. Niente di straordinario, ma "resuscitare" vecchie glorie rimaste silenti per anni e accorgersi che funzionano senza fare una piega con un sistema operativo libero dà una certa soddisfazione. E così, trovandoci una sera ogni tanto, nel giro di qualche anno abbiamo recuperato decine di macchine, ingombrando la sede del laboratorio di Padova e infestando i LinuxDay e i Webbit del PLUTO Padova con computer "liberati".

ReFUN come progetto PLUTO

Lo scorso anno la proposta di estendere questa esperienza a tutto il PLUTO ha fatto nascere ReFUN come progetto nazionale: è un modello su come svolgere l'attività di recupero in modo tecnicamente sicuro e nello stesso tempo rispettoso della storia: installare SO liberi non significa voler condannare all'oblio sistemi operativi che possono essere di gran pregio anche se non liberi. Con tutti i limiti del paragone, non vogliamo mica fare come i taliban afghani, che spararono sulle statue millenarie dei buddha solo perché non apprezzavano una religione diversa...

Il progetto ReFUN è descritto nelle pagine di http://refun.pluto.it, perciò è inutile ripetersi. A differenza di altri pregevoli e lodevoli progetti di retrocomputing, come quelli del Golem di Empoli, del Faber di Padova e di altri che non conosco, ReFUN non ha uno scopo dichiaratamente orientato al sociale (recupero di hardware di seconda mano con GNU/Linux per offrirlo a chi non può permettersi un PC nuovo), ma ha come obiettivo più squisitamente hackereccio quello di valorizzare il vecchio hardware di qualità divertendosi e scoprendo cose nuove. Se il frutto del lavoro hackereccio serve a qualcuno e rende il mondo migliore, meglio!

Oltre a darci soddisfazione personale (ogni macchina diversa è una sfida), questa attività ci ha dato l'opportunità di scoprire soluzioni originali ed estremamente efficaci che sono state poi abbandonate, come l'incredibile IndyCAM, le sospensioni in gomma dei dischi del microVax 3300, il cavo unico per video, tastiera, mouse e audio del NeXT o la straordinaria integrazione delle Sparcstation IPX (che nonostante l'impaccamento lasciava lo spazio sulla motherboard per l'immagine del gatto, "mascotte" del progetto). Inoltre ci siamo spesso chiesti perché i moderni PC si ostinino ad usare dei BIOS così rozzi quando i programmi di monitor di Digital, SUN e HP di 10 anni fa fanno molto di più e molto meglio, e come mai l'input/ouput di disco pesi meno sulle vecchie macchine lente che sui nostri PC desktop. Come vedete c'è da riflettere e da imparare.

Il documento presentato al LinuxDay del 2002

Con la pagina che segue abbiamo presentato nel LinuxDay del 2002 i nostri primi risultati e le nostre motivazioni. Dopo altri due anni di lavoro con questi rottami di qualità non saprei descrivere meglio questa attività

Perché lavorare su computer vecchi invece che su quelli ultramoderni e ultraveloci? Perché cercare di installarvi sistemi operativi free? Spesso i vecchi computer sono stati costruiti secondo criteri di qualità che è difficile riscontrare su quelli nuovi, per cui a parità di performance 'grezza' (megahertz di clock o velocità di trasferimen-

to) i loro componenti sono meglio integrati. I sistemi operativi liberi similmente hanno criteri costruttivi determinati dal piacere di fare una cosa fatta bene, non dalle scadenze commerciali del reparto vendite. Inoltre per noi hanno importanza gli aspetti estetici: i modelli che presentiamo oggi, le Sun SparcStation IPX e Next Cube, entrambe macchine della classe workstation, sono belle, cioè sono oggetti che ci piacerebbe molto avere sulla scrivania. Non solo per l'eleganza o la compattezza della scatola, ma per la funzionalità e la corretta integrazione dei loro componenti. Per quanto riguarda il software free, il piacere estetico è dato dalla consapevolezza che è stato scritto non solo perché funzioni bene, ma perché rappresenti chi lo ha scritto davanti ad una comunità di pari che possa riconoscerne eleganza e funzionalità. Questo si somma al piacere di dimostrare che si possono fare cose utili e belle non necessariamente in cambio di denaro.

I sistemi operativi free, come Linux e (Net/Free/Open)BSD ci permettono di divertirci ancora ad utilizzare computer che altrimenti sarebbero "obsoleti". Questo perché il software di qualità gira meglio su hardware di qualità ed il risultato può essere veramente sorprendente.

Il vecchio computer rappresenta una sfida: va pulito da anni di polvere e di uso, da lunghi periodi di immagazzinamento "selvaggio" (a volte alle intemperie, sotto i pini), richiede documentazione, ricerche, ha modalità di funzionamento diverse da quelle dei PC, svela tante strade che la ricerca ha percorso e che sono state abbandonate nonostante esprimano delle ottime idee, e alla fine offre la soddisfazione di una utilizzabilità a volte inaspettata da parte di un oggetto che altrimenti avrebbe solo appesantito il nostro bilancio ecologico come spazzatura; in cambio, è vero, dobbiamo essere più pazienti e meno esigenti.

*Tutte le macchine che presentiamo possono utilizzare il sistema operativo sviluppato per loro dalle case produttrici o uno o più sistemi operativi Free, come Linux o *BSD. Di solito i sistemi free sono più veloci...*



Riferimenti webografici

- *Progetto ReFUN:*

http://refun.pluto.it

- *PLUTO Project:*

http://www.pluto.it

- *Laboratorio PLUTO Padova:*

http://www.plutopadova.org

- *MUSI, Museo didattico di Storia dell'Informatica:*

http://www.fwtunesco.org/musi

- *CLAC, Comunità per le libere attività culturali:*

http://www.fwtunesco.org/clac

- *GOLEM:*

http://golem.linux.it

- *Progetto FABER:*

http://faber.linux.it

L'autore

Alberto Cammozzo, alias mmzz, fa il computer-sitter presso un'università italiana.

Ha imparato l'uso di cd, ls e poi anche su su un AT&T 3B2 ai tempi del BBS del DEI dal quale il PLUTO è nato. Il BBS ora è sparito, e quello che è peggio ne è sparita anche la memoria. Passando per Minix su Atari è approdato a Linux, ma sogna ancora un computer che faccia "Poof".

È responsabile del progetto ReFUN, ma chi fa il grosso del lavoro è Danilo Selvestrel (danny[AT] pluto.it).

[a cura di Tn]

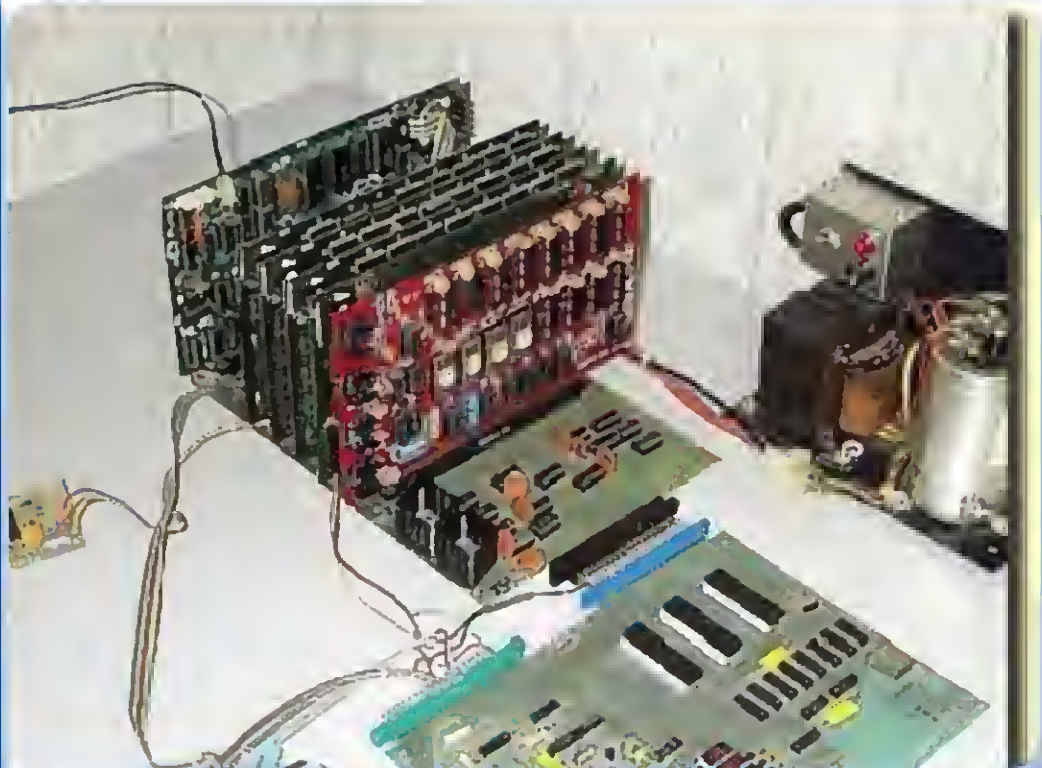
In queste pagine le foto di alcuni significativi sistemi recuperati



Le prove di Jurassic News

MOS Technology -- KIM-1

Il sistema di sviluppo single board KIM-1 della MOS Technology è stata una delle prime piastre a microprocessore sulla quale hanno potuto mettere le mani gli appassionati di elettronica digitale e imparare così i primi rudimenti (e non solo quelli).



Il KIM-1 in configurazione "espansa" fino a diventare un calcolatore personale completo.

Contesto storico

Le aziende produttrici di micro processori avevano un grosso problema da risolvere: allargare il mercato abilitando tecnici e progettisti all'utilizzo di un micro computer all'interno dei loro progetti.

Obiettivo tutt'altro che facile perché come ogni nuova tecnologia, quella dei micro era guardata con un certo sospetto: troppo integrata, troppo potente e alla fine anche troppo complessa da utilizzare. I sistemi di sviluppo single-board, come appunto il KIM-1, furono costruiti proprio a scopo educational, anche se alla fine crebbero fino alla dimensione del personal computer.

Una di queste aziende era la MOS Tecnology, produttore del chip 6502 che nel 1976 annunciò il rilascio del KIM-1 Microcomputer System. Esso consisteva in una board con CPU a 1 MHz di clock, 1 Kbyte di RAM e 2 Kbyte di ROM con una unità di I/O costituita da una tastiera "stile calcolatrice" e una fila di sei display a sette segmenti come output. Il costo era di 245 dollari, sufficientemente basso per permetterne l'acquisto anche a singoli appassionati, oltre alle aziende che si promettevano di progettare qualcosa attorno al chip

programmabile.

Il business della MOS Tecnology e del suo CEO Chuck Peddle, era quello di costruire chip elettronici e lo stesso proprietario aveva lavorato personalmente alla stesura del progetto mp6502. Come spesso è accaduto negli States, uno spin-off viene presto nel mirino di potenze economiche interessate all'acquisizione delle tecnologie, mentre i primi ideatori, che sono spesso anche i proprietari delle ditte, non hanno quella capacità imprenditoriale (e nemmeno i capitali) per allargare i loro orizzonti verso i mercati nazionali ed internazionali.

Così quasi subito dopo l'annuncio del KIM-1, la Commodore si compra la MOS Technlology e si lancia nel mercato dell'home computer portandovi quei personal che sono noti a tutti: PET, C64 e Amiga, per citare le macchine più conosciute.

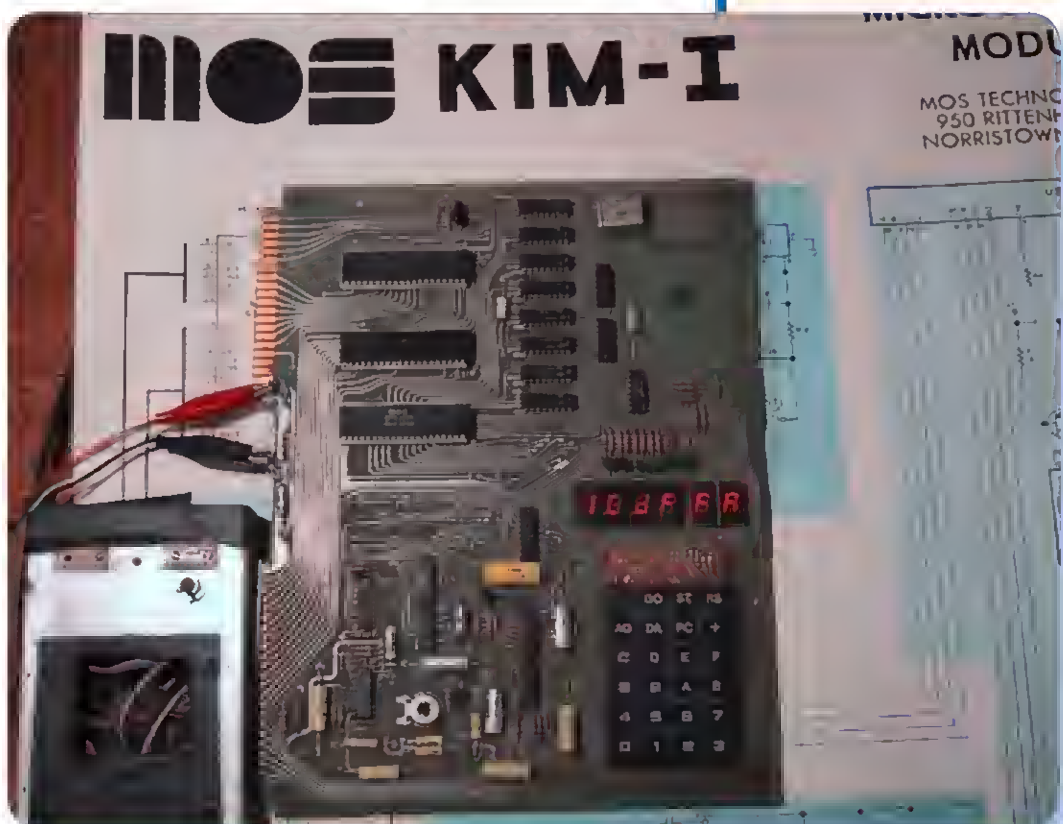
Ma come mai la Commodore aveva puntato l'azienda di Peddle? Chi era Commodore e cosa ci stava a fare nel mercato high-tech americano?

La Commodore Computer, come sarà poi chiamata, nasce nel 1954 ad opera di Jack Tramiel con lo scopo di vendere ed aggiustare macchine da scrivere e calcolatrici. Qualche anno dopo troviamo Jack in Canada al comando di un piccolo impero per la fornitura di macchine da ufficio a basso costo. Qualche anno più tardi la ditta tenta la strada della produzione di macchine calcolatrici elettroniche ma viene di fatto surclassata dal colosso Texas Instruments che poteva vantare una tecnologia inarrivabile per il buon Jack Tramien.

L'occasione per l'acquisto della MOS Technology fu l'ancora di salvataggio per una azienda che era su un binario morto e che avrebbe finito per diventare semplicemente un fornitore di soluzioni ma mai un produttore, e all'epoca i soldi si facevano vendendo hardware!

Non è dato sapere quanto sia costata l'operazione MOS Technology a Tramien, ma con essa, e grazie al fatto che si prese in sovrappiù anche lo stesso Peddle, che rivelò tutte le sue grandi doti

Il sistema della MOS Technology prevede hardware e manuali di istruzione per imparare tutto sui microprocessori.



di progettista proprio alla Commodore, e che già nei primi mesi del 1977 riusciva a presentare il suo prototipo del PET, il primo personal computer basato sul 6502 con monitor integrato e registratore a cassette.

Le doti commerciali di Tramien furono decisive quando riuscì a coinvolgere la catena di negozi Radio Shack nella vendita del PET. Il calcolatore personale era offerto per la prima volta in un negozio di articoli di elettronica, assieme alle radio a transistor e ai frullatori.

Il primo sistema commerciale di Commodore fu denominato PET 2001 e costava circa 600 dollari USA.

Da questi primi passi un increscendo continuo di espansione del listino prodotti e del business a livello mondiale con l'incredibile cavalcata durata per tutti gli anni '80.

Il microprocessore 6502 ebbe in Commodore la sua ultima applicazione nella serie 64 e derivati, mentre poi con Amiga la ditta scelse il salto tecnologico ai 32 bit.

Eravamo di fatto ancora negli anni '80, ma la tecnologia correva veloce e dall'annuncio dell'Amiga 1000 (1985) alla definitiva liquidazione della società passarono pochi anni: fino al 1994.

Il 1976 fu un anno di grande fermento per l'elettronica digitale e per il microprocessore 6502. Infatti nello stesso anno anche la Rockwell International (fabbricante di 6502 su licenza), assieme a Synertek rilasciò il proprio kit di sviluppo: l'AIM 65. L'AIM comprendeva una tastiera ASCII, uno display LED di 20 caratteri alfanumerici a 14 segmenti ed una piccola stampante da registratore di cassa. Anche l'altro sourcer, la Synertek, progettò e mise in commercio il proprio Kit, chiamandolo con ovvia scopiazzatura SYM-1. Esso era un miglioramento rispetto al KIM-1, di fatto minimale ma non raggiungeva la complessità e il costo dell'AIM65, davvero oltre la portata del normale hobbista.

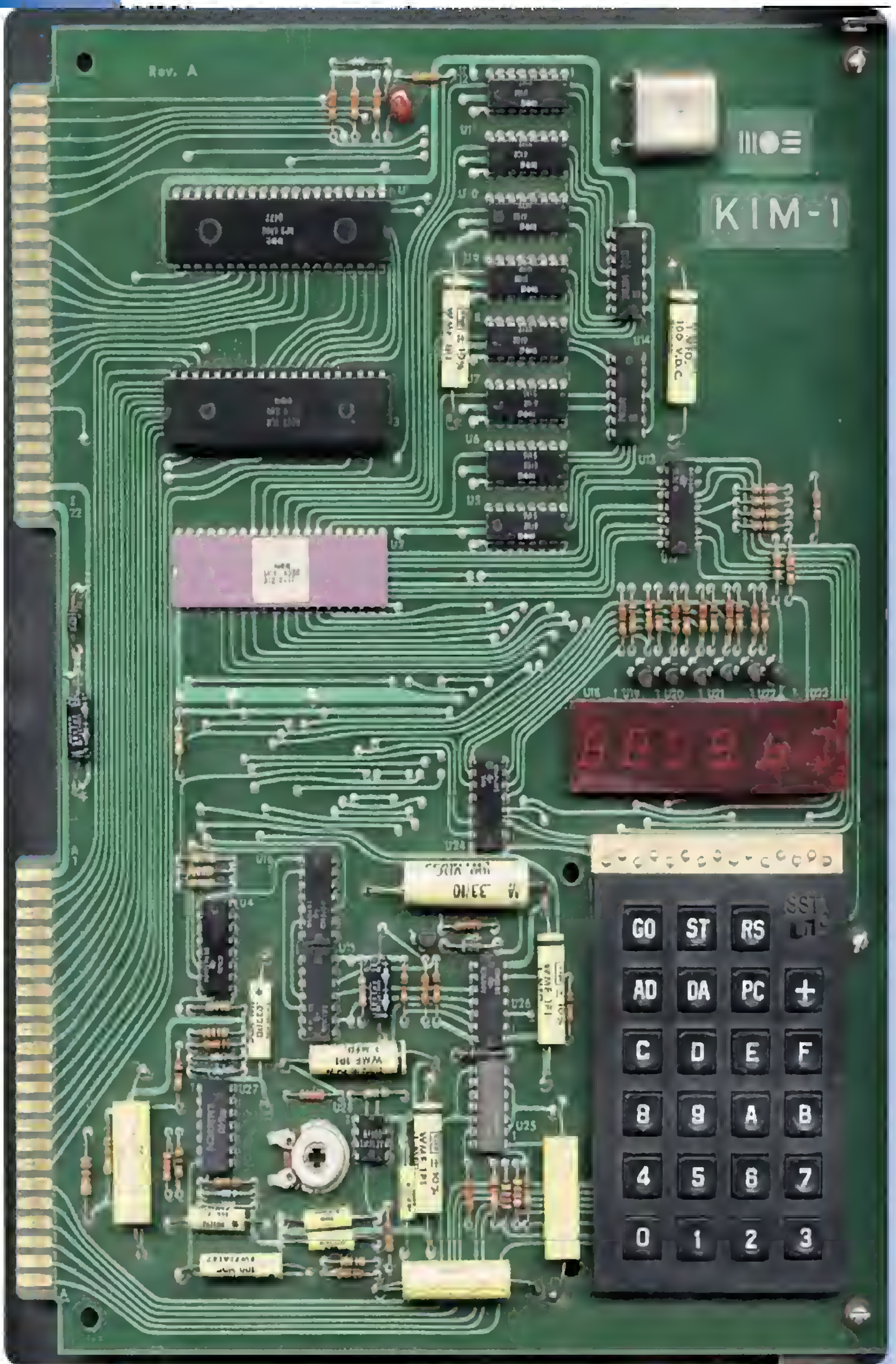
Primo approccio

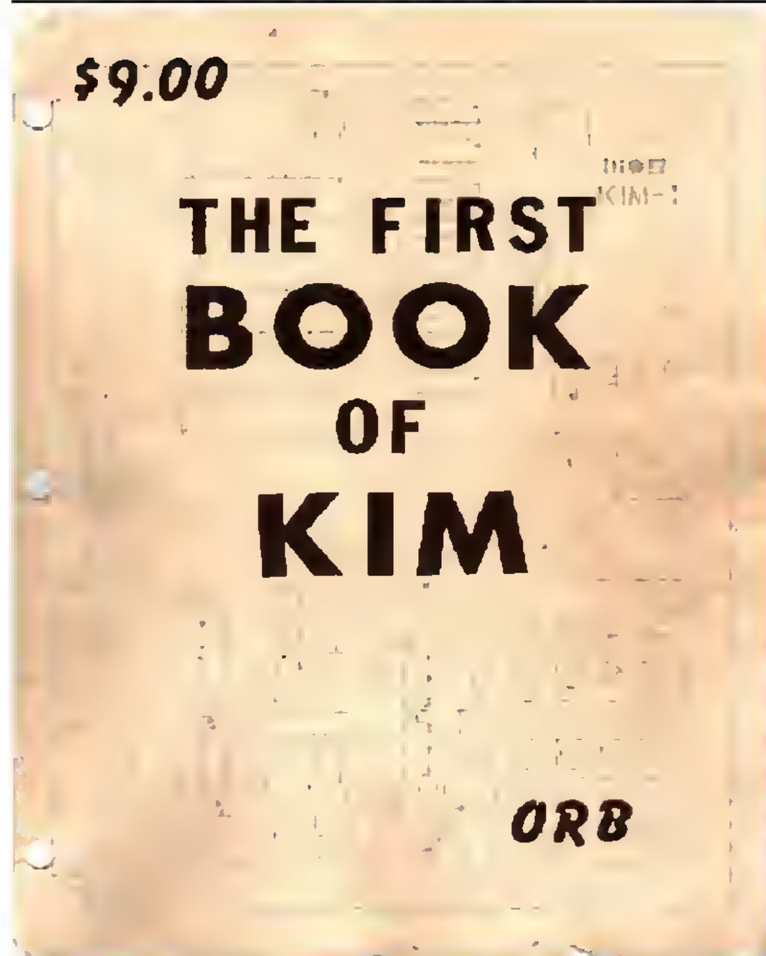
La piastra è corredata da due manuali: l'hardware guide e lo user's guide. Sul primo documento si trova una ampia descrizione di tutte le componenti hardware, la mappatura delle piedinature, etc...; nello user's guide il programmatore alle prime armi troverà un corso passo-passo per entrare nel fantastico mondo della programmazione in linguaggio macchina.

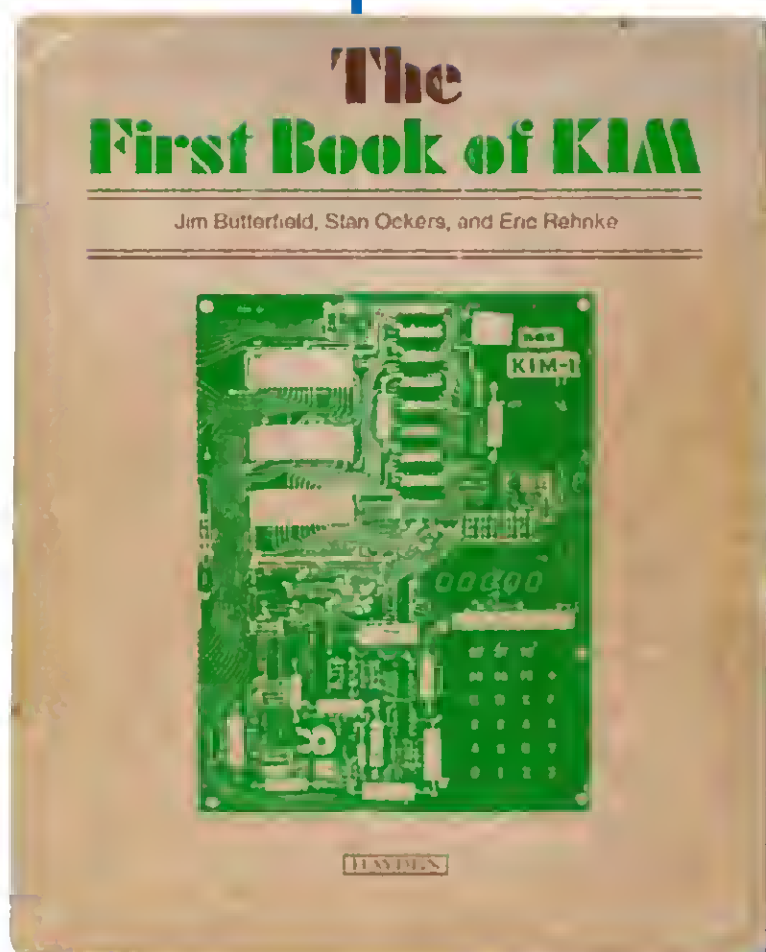
Usando il sistema come supporto all'apprendimento, non è necessaria nessuna espansione. Il computer prevede infatti tutto il necessario nell'unica piastra a doppia faccia, display e tastiera compresi.

La disposizione di tastiera e display obbligano l'utilizzo della piastra appoggiata su un piano di lavoro, ovviamente isolato, con sulla sinistra il pettine di espansione che rimarrà per forza di cose inutilizzato, tolte le connesioni di alimentazione ed eventualmente per il registratore a cassette.

Nella pagina a fianco una riproduzione quasi a grandezza naturale della scheda KIM-1. La scheda mostrata è la prima versione costruita direttamente da MOS Technology.


Rev. A
KIM-1
GO
ST
RS
SST
AD
DA
PC
+
C
D
E
F
8
9
A
B
4
5
6
7
0
1
2
3





L'aspetto del sistema appare quantomeno precario e soprattutto "esposto", nel senso che i chip, così "al vivo" potrebbero essere danneggiati da qualche manovra errata, come la caduta accidentale di un cacciavite, materiale certamente presente in un laboratorio di elettronica, ambiente d'elezione per questo prodotto.

Questo utilizzo immediato e "spartano" del sistema di sviluppo, potrebbe crescere facilmente con l'aggiunta di espansioni come tastiera teletype e terminale video. In questo caso la mother board diventa la scheda microprocessore di un assemblaggio che ha come base un bus di collegamento.

E' facile distinguere sulla piastra i vari chip e blocchi funzionali: il microprocessore in alto a sinistra e i due chip 6530 subito sotto. Sulla destra di questi il banco di RAM formato da otto chip da un kilopit ciascuno.

Il gruppo display a sette segmenti, pilotato da uno dei chip 6530 con l'ausilio di sei transistor è posizionato sopra la tastiera.

La tastiera assomiglia a quella di una calcolatrice anche se le serigrafie sui tasti non hanno la disposizione standard per le cifre e mancano ovviamente i simboli per le operazioni aritmetiche ad eccezione del +, che però nell'utilizzo ha un'altro significato. I tasti operativi comprendono le sedici cifre della notazione esadecimale (da 0 a F), sette tasti funzionali e uno switch che consente di commutare il funzionamento in single step.

I tasti funzionali servono a richiamare la visualizzazione dei registri, a confermare l'immissione dei dati e a lanciare il programma preparato in memoria.

A fianco della tastiera, sulla sinistra, trova posto una zona occupata da elementi discreti fra i quali spiccano i numerosi condensatori e un trimmer di regolazione che serve per variare la frequenza di campionamento per il segnale del registratore a cassette.

Il profilo piuttosto appiattito della tastiera e dei display a sette segmenti, ne rende difficile approntare la costruzione di un contenitore che possa proteggere la circuiteria lasciando nel contempo libera la digitazione.

Diverso il caso in cui il sistema sia fatto crescere fino alle dimensioni di un "vero" personal, con bus di espansione, schede aggiuntive, terminale video, etc...

Le varie versioni rilasciate prima dalla MOS Technology e poi dalla Commodore, si distinguono da pochi particolari e dalla modifica del logo che è stampigliato in alto a destra accanto al quarzo utilizzato come riferimento per il clock.

Hardware

Il microprocessore utilizzato è la versione a 1 MHz del micro 6502 che equipaggerà dopo il KIM-1 numerosi altri home di successo degli anni '80.

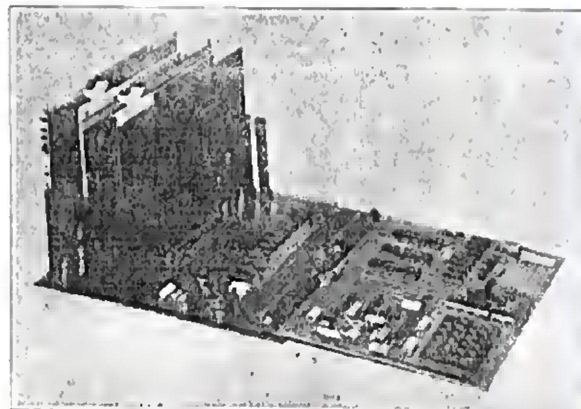
Sulla piastra base sono installati 1 Kbyte di RAM e 2 Kbyte di ROM contenente un monitor che consente una rudimentale gestione e programmazione del sistema.

La tastiera, di chiara derivazione calcolatrice-sca è una matrice di 6 righe di quattro tasti per un totale di 23 pulsanti e uno switch che funge da interruttore per l'attivazione della modalità single step. I display a sette segmenti che formano l'unità di uscita sono organizzati sopra la tastiera e costituiscono una riga di sei elementi, sufficienti per contenere la codifica esadecimale dell'indirizzo (quattro display) e relativo contenuto di memoria (due display rimanenti).

Oltre al microprocessore il circuito prevede due grossi chip che sono dei 6530 per il controllo dell'I/O attraverso quattro porte da 8 bit programmabili. Interessanti le possibilità di espansione con un doppio connettore di uscita a pettine da 22 pin ciascuno in doppia faccia (in pratica ci sono 88 pin di collegamento con l'esterno.







Documentazione e brossure erano ricercatissimi. Sotto il faccione di Tramiel, ceo di Commodore Computer, uno dei protagonisti indiscussi dell'informatica presonale negli anni '80



MOS
KIM-1
microcomputer
system

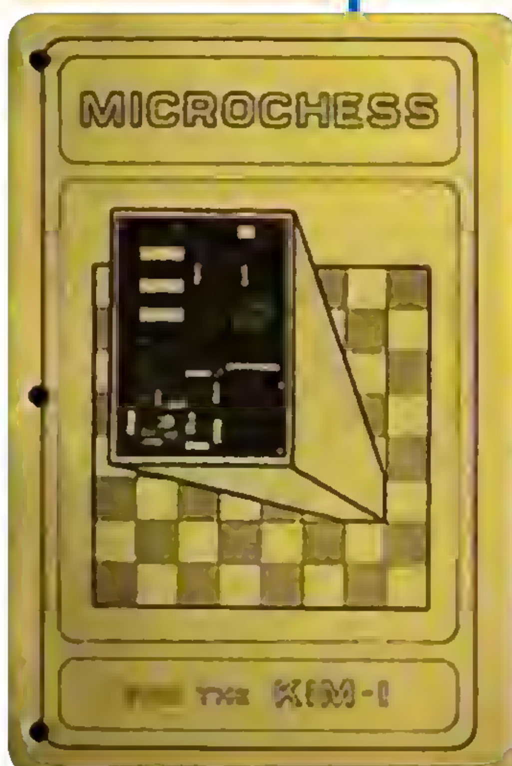
• A COMPLETE MICROCOMPUTER
• ONLY $245
• NOT A KIT!
• FULLY ASSEMBLED
• FULLY TESTED
• FULLY WARRANTED
• OPERATES WITH
• KEYBOARD & DISPLAY
• AUDIO CASSETTE
• TTY
• KIM-1 INCLUDES
• HARDWARE
KIM-1 MODULE WITH
6502 μP ARRAY
6530 ARRAY (2)
1 K BYTE RAM
15 I/O PINS
• SOFTWARE
MONITOR PROGRAMS (STORED IN 2048 ROM BYTES)
• FULL DOCUMENTATION
KIM-1 USER MANUAL
SYSTEM SCHEMATIC
6500 HARDWARE MANUAL
6500 PROGRAMMING MANUAL
6500 PROGRAMMER'S REFERENCE CARD

USE THIS FORM TO ORDER YOUR KIM-1 TODAY!

MOS TECHNOLOGY, INC.
KIM-1, 950 Rittenhouse Rd.
Norristown, PA 19401

www.commodore.ca

(Sopra)
La pubblicità del sistema single board su una rivista di elettronica.
(Sotto)
La distribuzione del software "fai da te" con il listato da digitare, un po' come uno spartito musicale.



I due chip 6530 che coadiuvano la CPU sono due interessanti componenti che contengono la logica di codifica di indirizzi per il controllo delle periferiche, due porte bufferizzate da 8 bi in ciascuno dei chip e nel caso del 6530 anche un Kbyte di ROM e 64 bit di RAM indirizzabili dal bus.

L'alimentazione è esterna e consiste in due tensioni di 5 Volt e +12 Volt, rispettivamente da 1.2 e 0.1 Ampere.

Le connessioni I/O predisposte e supportate sono verso la tastiera/display, verso l'interfaccia a cassette audio e verso una periferica di tipo teletype sia in current-loop che in seriale (con qualche adattamento).

La mappa della memoria prevede il primo K di RAM posizionato in basso, cioè da 0x0000 a 0x03ff. I banchi di RAM, la ROM e gli altri indirizzi di I/O supportati dai due chip 6530, sono mappati a partire da 0x1700.

L'espansione hardware del sistema è lasciata alla buona volontà dell'utente e ai numerosi suggerimenti che ad essi pervengono via riviste e fanzine varie che non sono mancate. Del resto non mancano le informazioni hardware sul manuale a correndo e un hobbista dell'epoca era soprattutto, se non esclusivamente, un appassionato di elettronica.

Come supporto all'espansione del sistema la piastra madre è provvista di due connettori a pettine denominati "Expansion Connector" e "Application Connector".

La MOS Technology ha rilasciato anche una piastra di espansione chiamata KIM-4 che provvede sei connettori a pettine per altrettante espansioni e un connettore primario nel quale infilare la piastra CPU. La Kim-4 prevede anche un potenziamento dell'alimentazione e una bufferizzazione dei segnali che circolano fra le schede.

Uso

La prima cosa di cui occuparsi è naturalmente l'alimentazione. La MOS consiglia e descrive sul manuale hardware un esempio di circuito in grado di erogare le due tensioni necessarie (+5 e + 12 Volt).

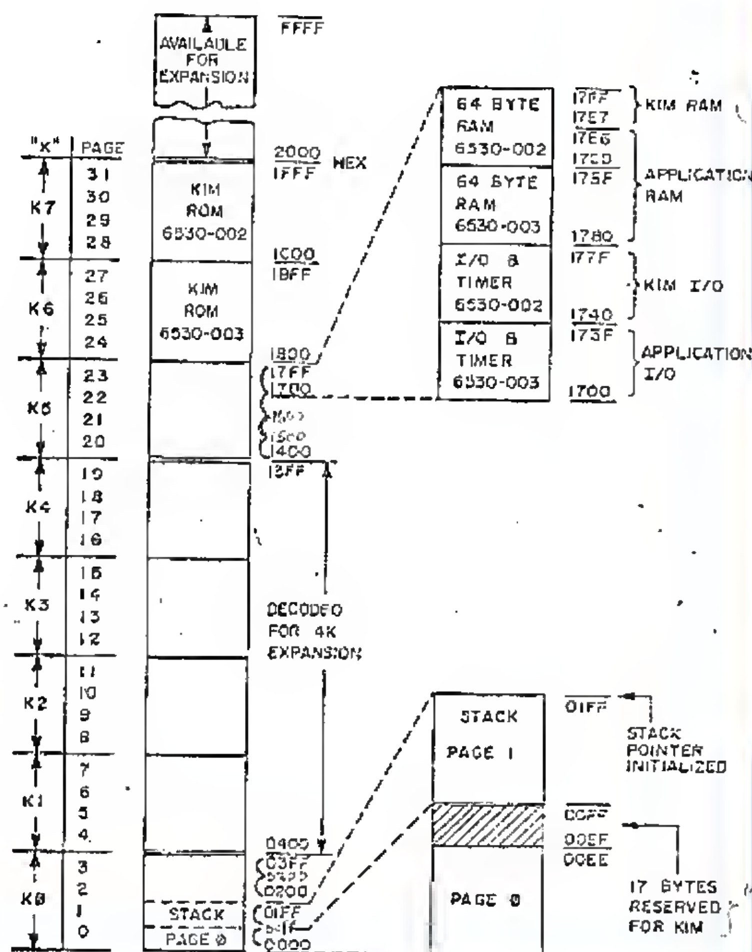
Collegata l'alimentazione è op-

portuno premere il tasto RESET sulla tastiera (è siglato RS) il quale reinizializza il sistema e rende il controllo al programma monitor ospitato dalla ROM di sistema.

Utile se non proprio indispensabile l'unità a cassette, anche questa non fornita in dotazione ma la casa costruttrice assicura il buon funzionamento di qualunque tape recorder anche a basso costo. Sulla porta di espansione "Application Connector" i pin A e P sono riservati all'output, mentre il pin L all'input dal registratore. Il Kim-1 prevede due segnali di uscita verso il registratore: uno a basso voltaggio (15 mVpp) e un'altro ad alto voltaggio (1 Vpp), quest'ultimo da usare se si desidera approntare una uscita audio di storage maggiormente sofisticata.

Se si desidera installare una teletype come unità di input/output al posto del tastierino/display presente sulla piastra, il KIM-1 mette a disposizione una connessione a quattro linee con protocollo Current Loop di corrente a 20 mA.

In questo caso vanno ponticellati due pin sul connettore Application per rendere edotto il sistema della presenza della periferica TTY. La MOS consiglia di installare uno switch al posto del ponticello per avere la possibilità di scegliere fra TTY e Keyboard interna senza ricorrere al saldatore. Se il terminale TTY è installato, la sequenza di boot del monitor (chiamato Operating Program) prevede il test della periferica per stabilire la velocità di comunicazione e la disabilitazione del tastierino esadecimale.

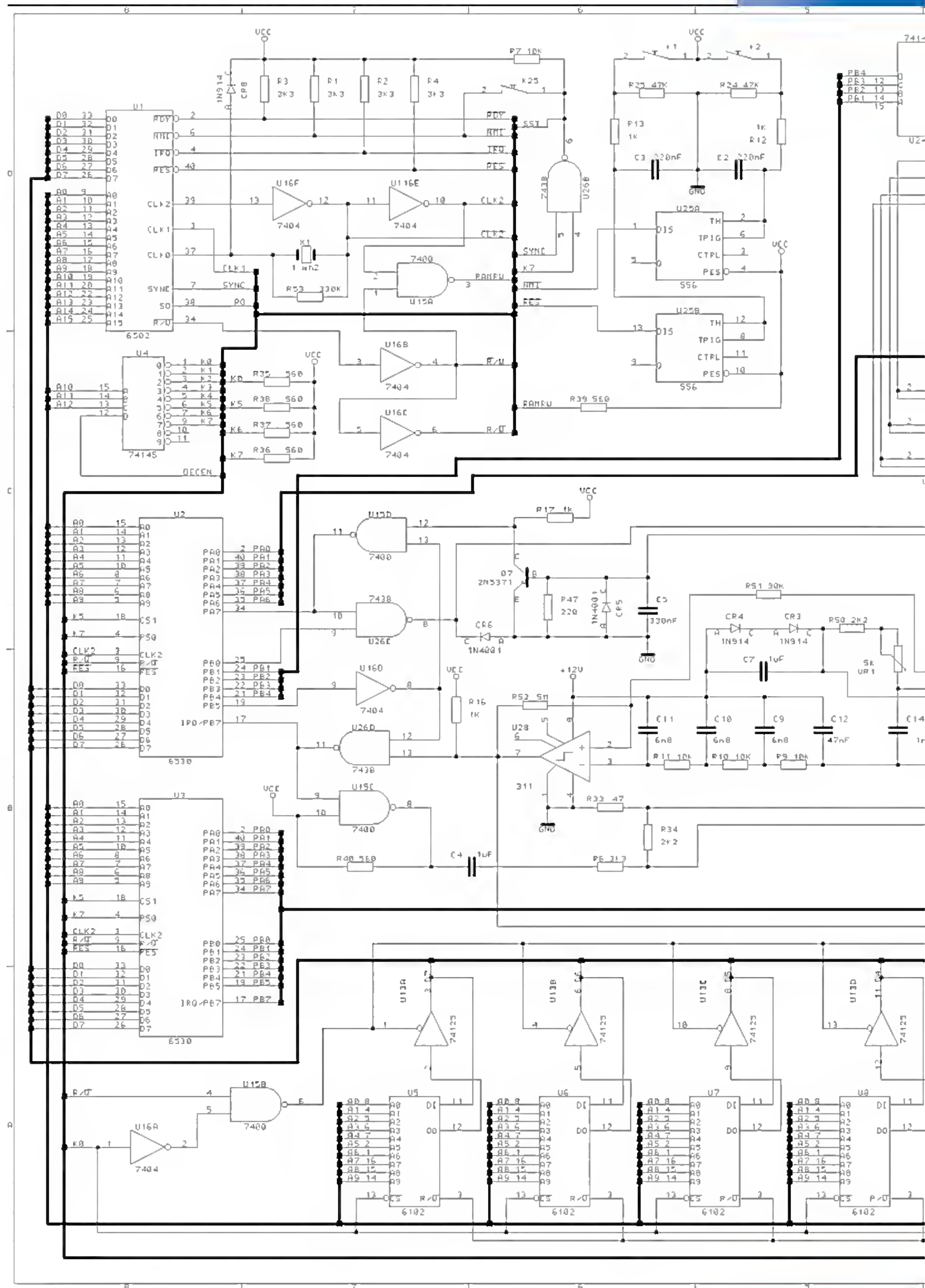


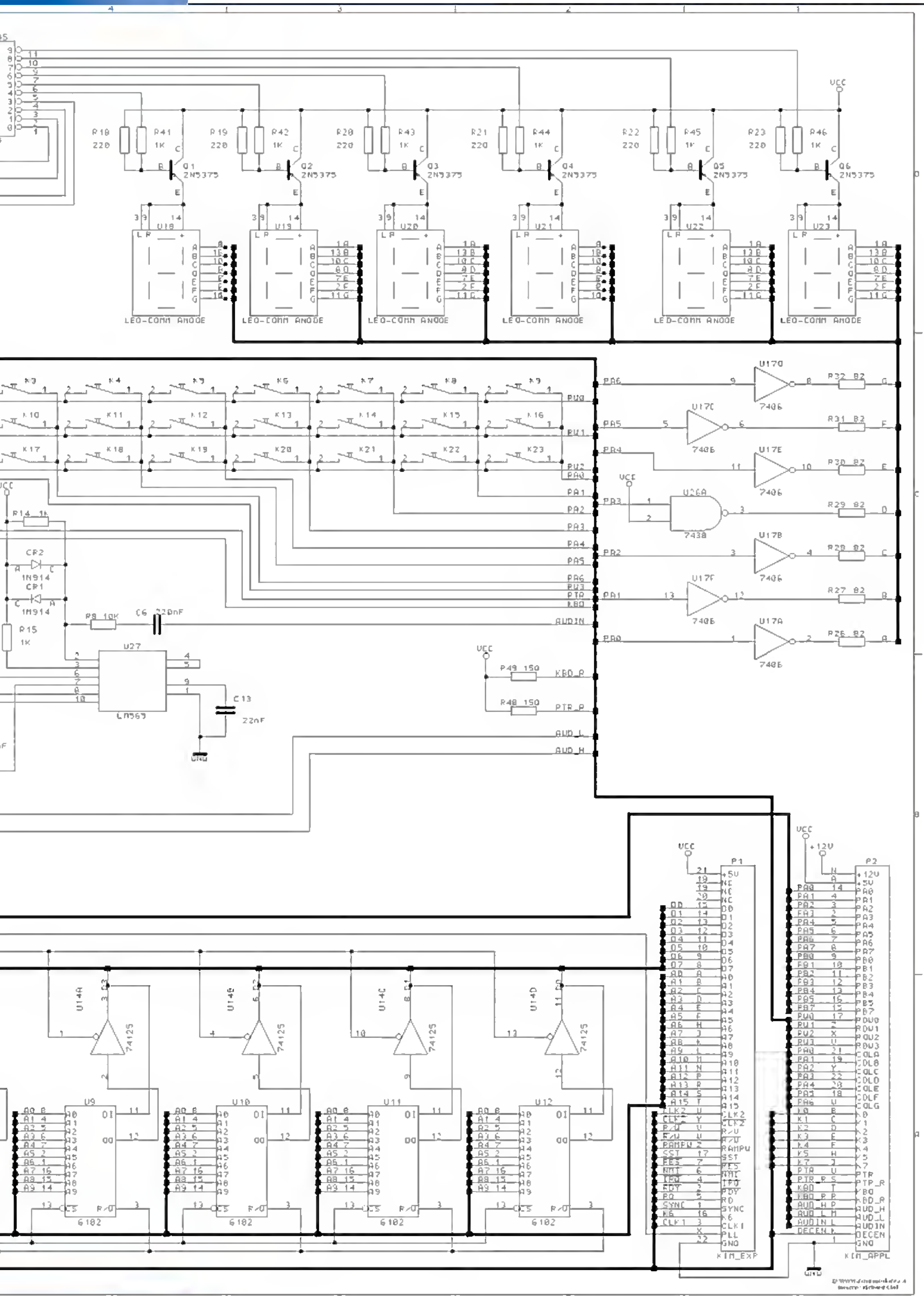
La mappa di memoria del sistema, recuperata da uno dei manuali ufficiali. Informazioni come questa erano essenziali per procedere nella programmazione della macchina.

Usando il tastierino esadecimale il monitor di sistema ha due modalità di funzionamento che si scelgono premendo AD (Address Mode) o DA (Data Mode).

Con AD i primi quattro display a sinistra mostrano l'indirizzo di memoria e il contenuto sui due display rimanenti a destra. Il tasto "+" muove al prossimo indirizzo.

In modalità DA le cifre premute sulla tastiera vanno a sostituire il contenuto della memopria all'indirizzo visualizzato; il tasto "+" in questo caso serve per passare al prossimo indirizzo confermando contem-





Kim-1
USER NOTES

JULY 1976 COMPLIMENTARY ISSUE

PAGE 1

MICROPROCESSOR SOFTWARE SEMINAR SCHEDULED FOR AUGUST 18-20

A three-day intensive course in software development, featuring the KIM-1, will be held at Turf Inn on Wolf Road in Colonie, New York. The course is being offered by three educators from Rensselaer Polytechnic Institute, Troy, New York.

"Hands-on" experience will be stressed as each student will receive his or her own KIM-1 and power supply. The cost of the seminar will be $495.00 complete with KIM-1, power supply, course notes, I/O interface, etc. or $275.00 if you already have a KIM-1.

Interested parties should contact J. C. Williams, R. K. MacCrone, or D. S. Yaney as soon as possible for registration or additional information about the course. They can be reached at (518) 270-6495.

They must have firm commitments by the end of July in order to insure that hardware will be ready by course time.

••••••••••••••••••••••

KIM-2, -3, AND -4 ARE ON THE WAY!!!

MOS TECHNOLOGY is now making more memory available for the KIM-1. Starting August 16, 1976, MOS will be shipping two new memory expansion boards--- the KIM-2 (4K static RAM) and the KIM-3 (8K static RAM). Both boards will be assembled, tested, guaranteed for 90 days, and fully burned-in with high-speed static RAM. All buffering and control logic will be included as well as on board regulators. A single KIM-2 or KIM-3 can be wired directly to the KIM-1, but, if you need more more memory, you'll have to wait for the KIM-4 Motherboard, which they say will be available shortly. The price? $179.00 + $5.00 (shipping) for the KIM-2, and $298.00 + $5.00 (shipping) for the KIM-3.

••••••••••••••••••••••

poraneamente l'eventuale modifica del contenuto della locazione di memoria.

Manca il tasto per fare un passo indietro, ma va ricordato che la progrrammazione in linguaggio macchina si effettua praticamente solo “in avanti” e la necessità di rivedere il contenuto in memoria può essere facilmente perseguito impostando in modo AD l'indirizzo di partenza e proseguendo di conseguenza.

L'esecuzione di un programma prevede di impostare il valore dell'indirizzo di partenza nel registro PC e di premere il tasto GO per procedere. Nell'eventualità che si fosse attivato il Single Step, il pulsante GO permette di effettuare l'esecuzione per singolo passo visualizzando indirizzo e corrispondente contenuto sul display mano a mano che l'esecuzione procede.

Un'altro tasto operativo utile è ST che è in pratica un break utile per interrompere un programma in loop e restituire il controllo al monitor.

Il KIM-1 non è dotato di un vero sistema operativo per cui anche semplici operazioni come storing e retriving di data dal nastro magnetico prevedono una fase di preparazione della memoria con opportuni dati e l'esecuzione di una specifica routine interna partendo da un ben determinato indirizzo.



THE SYM USERS' GROUP NEWSLETTER

VOLUME IV, NUMBER 2 (ISSUE NO. 16) - SUMMER 1983 (MAY/JUN/JUL/AUG)

SYM-PHYSIS is a thrice yearly publication of the SYM Users' Group, P. O. Box 319, Chico, CA 95927. SYM-PHYSIS and the SYM Users' Group (SUG) are in no way associated with Synertek Systems Corporation (SSC), and SSC has no responsibility for the contents of SYM-PHYSIS. SYM is a registered trademark of SSC. SYM-PHYSIS, from the Greek, means the state of growing together, to make grow, to bring forth.

We welcome for publication all articles dealing with any aspect of the SYM-1, and its very close relatives. Authors retain all commercial copyrights. Portions of SYM-PHYSIS may be reproduced by clubs and educational institutions, and adaptations of programs for other computers may be freely published, with full credit given and complimentary copies provided to SYM-PHYSIS and the original author(s). Please include a self-addressed stamped envelope with all correspondence.

Editor/Publisher: H. R. "Lux" Luxenberg
Business/Circulation: Jean Luxenberg
Office Staff: Joyce Arnovick, Sandy Bell

SUBSCRIPTION RATES: (Volume IV, 1983, Issues 15 - 17)

USA/Canada - $13.50 for a volume of three issues. Elsewhere - $14.00. Make checks payable in US dollars to "SYM Users' Group", P. O. Box 319, Chico, CA 95927, Telephone (916) 895-8751.

BACK ISSUES ARE AVAILABLE AS FOLLOWS:

Issues 3 through 6 (Volume I, 1979/80), are available for $12.00, US/Canada, and $15.00, First Class/Airmail, elsewhere.

Issues 7 through 10 (Volume II, 1981), are available for $10.50, US/Canada, and $14.00, First Class/Airmail, elsewhere.

Issues 11 through 14 (Volume III, 1982), are available for $10.50, US/Canada, and $14.00, First Class/Airmail, elsewhere.

ANOTHER OUTSTANDING OFFER TO THE SYM COMMUNITY

On the page to the right is a map of the environs of the Caves of Nirdarf. This map is part of the 28 page manual for "SYM-VENTURE", sent to us, along with a (KIM-speed) cassette containing the object code for the game, by Mark Ganis. The object of the "adventure" is to find the "treasures" (gold and pearls), preferably visiting every location during the quest, and to return home (i. e., to the house), safely, with them.

While we can't usually find the time for most computer games, we did make time for this one, for a number of very valid reasons:

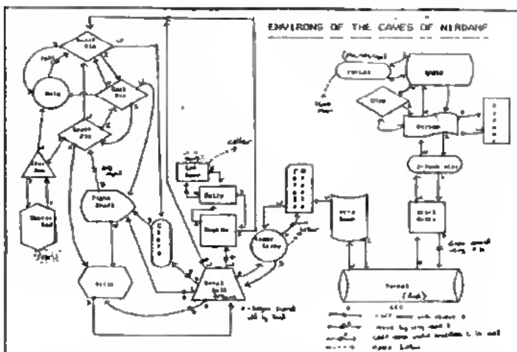
1. It can be played on a 2K SYM-1 WITHOUT a terminal!
2. The manual contains fully commented source code, as well as very well written instructions and the map, which will make playing the game much more fun, and definitely much less frustrating!
3. It is very inexpensive, almost at cost of media and shipping!
4. A study of the source code will not only reveal all the "secrets" of writing money-making Adventure-type games, but, because of the clever use of the 7-segment displays to display text messages, will also teach you much about how the SYM itself works!

SYM-PHYSIS 16- 1

SYM-VENTURE (c) is an adaptation (by permission) of Robert Leedom's original KIM-VENTURE (c), which we did not know existed, back in our KIM-1 days (incidentally, KIM-1 had only 1 1/8 K of RAM).

You may order either KIM-VENTURE or SYM-VENTURE (or both), cassette plus manual, at $15.00 each (overseas, please add $3.00 for Air Mail), directly from Mark Ganis, Sheridan Road, R. D. #3, Lebanon, NJ 08833. Copies may be duplicated by clubs, or users' groups for a $5.00 per copy royalty fee (very generous, and extremely reasonable, this!). Every SYMmer should send for a copy!

P. S. SYM-VENTURE, and "SWISS" CLOCK (see below), are ideal programs for demonstrating the potential of the simple (unexpanded) SYM-1!



MORE ON THE "SWISS" CLOCK

In the article "ADJUSTABLE REAL TIME (SWISS) CLOCK" (SYM-PHYSIS 13/14-91, we pointed out that the program was four bytes too long for an unexpanded SYM-1, or a SYM-2. The author himself, Mr. Schumacher, sent in a shortened version. We print, instead, the following postscript to a recent letter from Boris Goldowsky (author of the SYM-PHYSIS INDEX):

P. S. To fit "SWISS CLOCK" into 1K, use one of my favorite tricks: On lines 3400-3?75 & 3540-3550, replace ADC #$01 CLC with SEC. For astronomers, the clock can be made to show sidereal time by replacing 0302 and 03C1 with C2, 0305 with 0A, and 030D with C3.

SYM-PHYSIS 16- 2

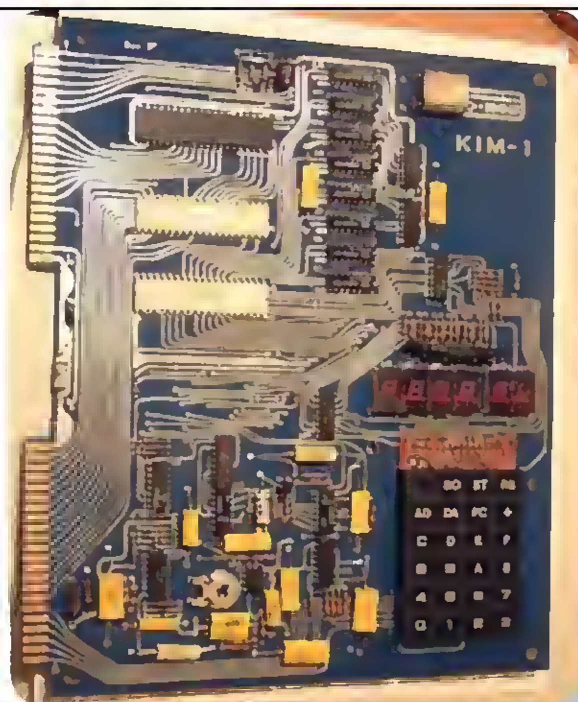
Software

La presenza di software, anche di una certa qualità (compatibile con le limitate possibilità elaborative della macchina) è stata assicurata principalmente da iniziative di tipo editoriale, bollettini più o meno tecnici e cose di questo genere.

Registriamo una collana di software in listato come ad esempio "Microchess", sviluppato appositamente per il KIM-1 o il porting dell'interprete TinyBASIC, offerto free dal suo curatore.

Per il resto non ci sono notizie che sia mai stato sviluppato un vero sistema operativo o software di particolare complessità; tutto questo in linea con il target del prodotto.

Sono numerosi comunque i listati reperibili qua e la e anche i progetti di espansione hardware o di controllo attraverso le porte di espansione collegate opportunamente con relè.



Conclusioni

Sicuramente chi ha potuto permettersi l'acquisto del sistema di sviluppo ha imparato praticamente tutto quello che c'era da sapere sul micro processore 6502 e sull'architettura degli home che sarà conservata più o meno invariata in tutti i progetti degli anni '80.

Certo non un sistema "chiavi in mano" e nemmeno troppo agevole da interfacciare, se non con una certa dose di capacità nell'usare il saldatore.

Il costo estremamente contenuto, possibile grazie all'essenzialità di questo prodotto, ha contribuito certo alle fortune successive del micro 6502 e grande fiuto ha dimostrato la Commodore a volersene appropriare in tempo per partecipare, anzi essere protagonista, della rivoluzione, forse la più importante, del ventesimo secolo.

[Sn]

(Pagina a fianco) Fanzine e bollettini usciti dalla geniale volontà di club per appassionati, forniscono una divertente fonte di informazioni di ogni genere sul sistema. Si va dalle prove, alla costruzione in proprio di espansioni, ai listati, alle notizie varie sul sistema.

(Sopra) Del Kim-1 ne esistono almeno quattro revision. Quella mostarta sopra è dovuta a Commodore. Si noti il diverso layout del tastierino con lo spostamento da destra a sinistra dello switch per il single step.

Biblioteca

Vintage Games

Le monografie vecchie e nuove che rappresentano una preziosa risorsa per chi ama il mondo dei computer in generale.

Scheda

Titolo:

Vintage Games

Sottotitolo:

An Insider Look at the History of Grand Theft Auto, Super Mario, and the Most Influential Games of All Time

Autore:

Bill Loguidice, Matt Barton

Editore:

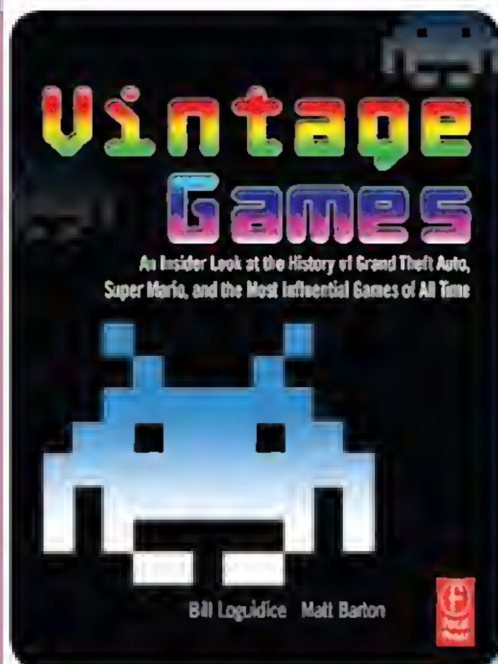
Focal Press

Anno: *2009*

Lingua: *Inglese*

ISBN:

978-0-240-81146-8



Abbiamo già avuto modo di apprezzare il lavoro di Matt Barton attraverso due suoi interventi apparsi sui Jurassic News [Bridges over trubled water; jn n. 3 e Games in Captivity; jn n. 5]. Ora Matt fa uscire, in collaborazione con Bill Loguidice (il nome è proprio questo, di evidenti origini italiane sebbene storpiato) per i tipi (si dice così?) della Focal Press, un volume monografico dedicato a quelli che gli autori giudicano "i videogiochi più influenti di tutti i tempi".

Gli autori stessi si rendono conto di quanto sia pericoloso includere un titolo nella rassegna e trascurarne dieci altri, per cui viene spiegata già nella prefazione la filosofia che ha portato alla realizzazione del volume e di conseguenza ad effettuare delle scelte.

Sono d'accordo con loro quando affermano che ognuno di noi avrebbe la sua propria classifica da promuovere in questo campo. Nè valgono le statistiche che si possono ricostruire partendo dalle liste "All fame" ospitate da tutti i siti che hanno come missione il retro computing. La scelta è stata fatta quindi sul concetto di "influenza", cioè cosa introduce di nuovo quel particolare titolo da punto di vista sia tecnico, di interazione o di argomento?

Rispondendo a questa domanda e forti di una rigorosa conoscenza che deriva dalla loro passione e (udite, udite) anche dal loro lavoro (sì, perché negli USA qualcuno fa di professione il ricercatore della cultura digitale, compresi i videogames!), i due autori presentano una lista di titoli ludici ordinati alfabeticamente.

Da "Alone in the Dark" fino a "Zork" trovano spazio 25 titoli che hanno influenzato (nei termini che si diceva) la cultura informatico-ludica nel periodo che va grossomodo dal 1978 al 1992.

La scelta di elencare i titoli in or-

dine alfabetico piuttosto che come ci si aspetterebbe in ordine cronologico, sta ad indicare la precisa volontà degli autori di non cadere nella facile classificazione per data e quindi nella inevitabile rassegna storica delle tecnologie disponibili sul mercato.

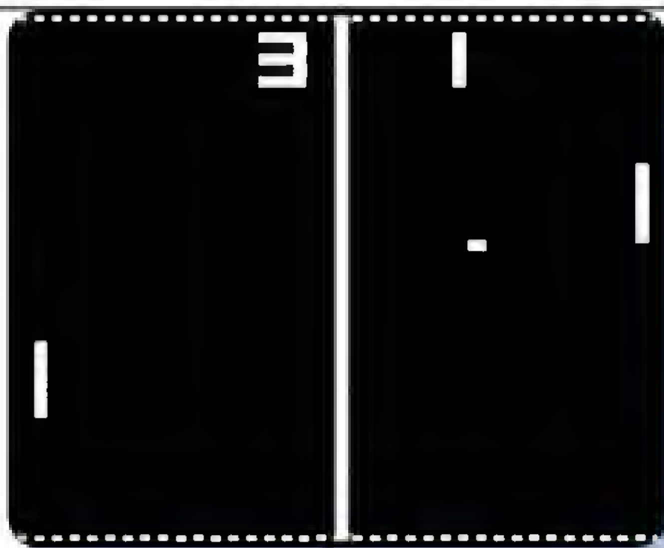
Pur elaborando venticinque capitoli, gli autori hanno superato anche la trappola del didascalismo, scegliendo non proprio un titolo per ogni lettera dell'alfabeto (ma quasi).

Leggendo o anche solamente sfogliando le 400 pagine del volume, si fa il classico "tuffo nel passato", anche grazie alle numerose immagini con screen ma anche sistemi usati per giocare (console e pc). Ma soprattutto si ha l'occasione di capire meglio come si colloca quel particolare game o filone, nel contesto di un comparto che è stato caratterizzato, e per certi versi lo è ancora, da una continua e incessante evoluzione tecnica sia nell'hardware che nel software.

Consci della vastità dell'argomento gli autori hanno realizzato anche una appendice virtuale al volume, cioè un sito Web dove sono disponibili screen shut aggiuntivi e articoli di una decina di giochi (fra i quali l'antesignano PONG) che non hanno trovato posto nella pubblicazione cartacea.

Se fanno sorridere o anche scrollare il capo, certi omini blocchettosi che si muovevano a scatti sul tv domestico, nel contempo ci si rende conto come ogni epoca produca manufatti che rispecchiano le capacità e gli strumenti che si hanno a disposizione. Un utile esercizio che concentrando questa evoluzione in soli quindici anni circa, ci aiuta a capire cosa significa "evoluzione" quasi in termini biologici: le mutazioni, la sopravvivenza del più adatto, la selezione sessuale, etc... (il 2009 è anche l'anno darwiniano, lo sapevate?).

[L2]



Da Pong (sopra) a Flight Simulator su un Commodore SX (sotto)



Nota dell'editore. Dopo aver consegnato l'articolo Lorenzo ci ha comunicato l'indirizzo di un progetto editoriale che si pone come obiettivo la traduzione in italiano del volume. Chi fosse interessato lo trova qui: http://www.vintagegames.it/index.shtml

Edicola

Commodore Free

In edicola o sul Web le riviste che parlano di computer, preferibilmente retro o free

Scheda

Titolo: *Commodore Free*

Sottotitolo: *A free to download Magazine dedicated to Commodore computers*

Web: www.commodore-free.com

Lingua: *Inglese*

Prezzo: *Free*

Primo numero: *Ottobre 2006*



Uno dei fenomeni più strani, se vogliamo, che sono stati favoriti da Internet è il fatto che le vecchie piattaforme home sono rimaste vive e vegete, ad onta degli anni che sono passati, e sono passati eccome!

Certo se non ci fosse stata la Rete non si sarebbero conservate le iniziative di mantenimento in vita di quelle macchine che sono state la passione di migliaia se non milioni di utenti che si sono avvicinati con esse per la prima volta al calcolatore elettronico. Senza la grande Rete delle Reti, rimarrebbero probabilmente pochissimi gruppi di appassionati sopravviventi ma in agonia, raccolti in club "amici di …" fino alla morte fisica dei soci, visto che il ricambio dei protagonisti è quantomeno fenomeno raro. Invece Internet ha fatto in modo che pur rimanendo in percentuale risibile rispetto ai fasti dei tempi migliori, i vari amanti dell'informatica personale prima maniera si sono aggregati in gruppi sovra-nazionali o quantomeno nazionali, dando origine al fenomeno dei siti dedicati alle piattaforme singole.

Il Commodore 64 è stata una di queste macchine gloriose e come tale ha generato la scia di appassionati mai stanchi di rinverdire gli antichi fasti convinti come sono di avere avuto fra le mani la macchina migliore possibile per l'epoca (e non solo).

Non è nostra intenzione lanciarci in diatribe così trite e ritrite su quale sia stato l'home migliore fra C64, Spectrum o Apple, ci basta considerarli ognuno al pari dei concorrenti per parlare con entusiasmo di quelle iniziative (editoriali in questo caso) che utilizzano il Web come vetrina e canale distributivo per mantenere in vita una passione.

All'indirizzo Internet http://www.commodorefree.com troviamo una rivista free, cioè liberamente scaricabile, che è dedicata appunto alla piattaforma Commodore (e al C64

in particolare).

La rivista mensile, redatta da un gruppo di volontari appassionati, si chiama "Commodore Free", è in lingua inglese ed è arrivata finora (nel momento in cui scriviamo) al numero 30. Il primo numero (secondo i miei calcoli) dovrebbe risalire all'ottobre 2006; attualmente sono in linea i numeri dalla issue 4 in poi , mentre purtroppo i primi quattro numeri sono stati rimossi -si legge sul sito- per ragioni di copyrights.

Il contenuto della rivista è a "tutto tondo", come si direbbe. Hardware, software (non solo giochi), news e altro confezionati per la lettura piacevole di coloro che si interessano di retro computing e delle macchine Commodore in particolare. Molto interessante la storia dei sistemi Commodore che inizia proprio da questo numero 4, ma altrettanto interessanti le interviste ai vari personaggi che ruotano attorno alla scena Commodoriana (o che sono comunque transitati per essa). Una menzione particolare poi ai progetti hardware che la rivista supporta, con lo scopo di dotare le macchine vetuste di quelle tecnologie che appaiono oggi indispensabili: interfacce IDE per storage su disco, scheda VGA per una uscita video compatibile con i monitor per PC, etc…

Non ci sembra invece particolarmente curato l'aspetto grafico, anche se negli ultimi fascicoli la cosa è migliorata in maniera notevole. La scelta "minimalista" crediamo sia suggerita dal fatto che la rivista è formattata per vari reader, compreso il formato html.

Allegato alla rivista anche un CD-ROM virtuale in formato D64. Ottima l'idea...

Conclusioni

Un plauso agli amici del sito www.commodorefree.com per l'iniziativa dal contenuto informativo molto elevato e per avere reso free questo loro sforzo. Senza ombra di dubbio un sito da visitare mensilmente per coloro che amano in particolare le macchine della casa canadese, ma anche il semplice appassionato di computer vintage potrà trovarvi spunti e un sacco di informazioni.

[Sn]





Emulazione

Soft6502

I mondi virtuali a volte possono essere molto realistici...

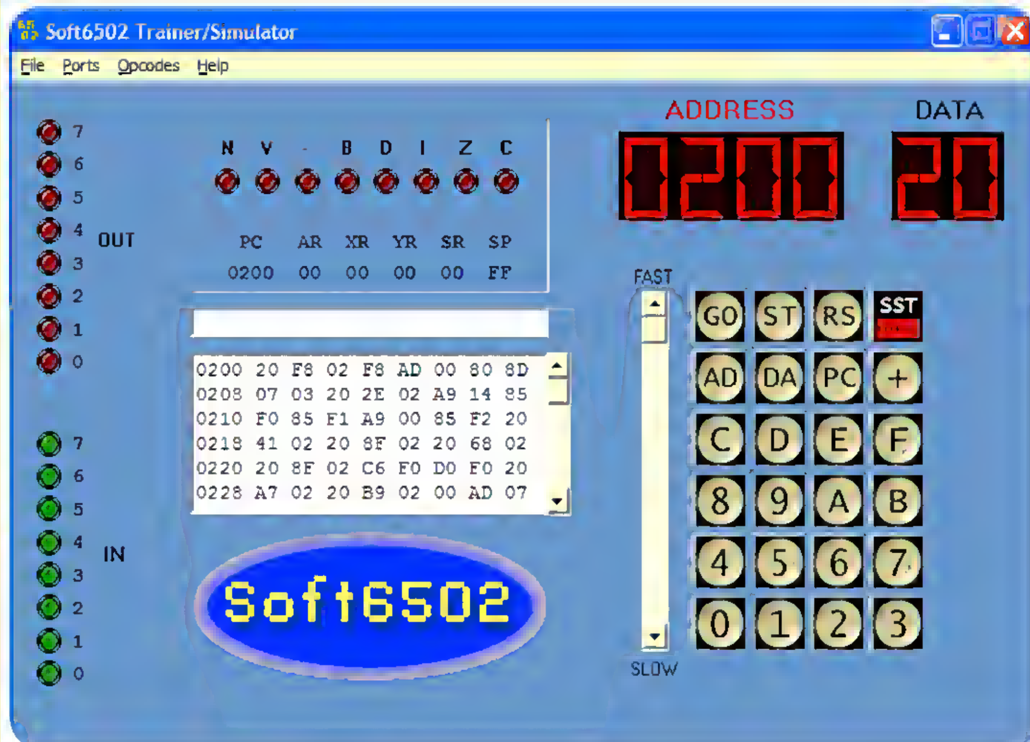


Figura 1.
Come si presenta il programma appena lanciato.

In occasione della prova del micro KIM-1, la redazione mi ha chiesto di scrivere su un emulatore per la piattaforma di sviluppo della MOS Technology (poi di Commodore). Cercando in rete ho scoperto che tutto sommato gli emulatori disponibili non sono poi molti (giusto tre o quattro, almeno da una prima ricerca superficiale). Uno di essi si basa sulla piattaforma MESS, ma volendo cercare qualcosa di "nativo" mi sono rivolto a questo software per Windows chiamato "Soft6502" e che è definito "Trainer/Simulator" dal suo autore, un certo C. Bond.

Devo confessare di essere rimasto un po' sorpreso di non trovarne a decine di questi emulatori: in fondo è una piattaforma molto semplice e manca del tutto il display video e periferiche varie che complicano non poco la vita degli sviluppatori di emulatori.

Soft6502 è una applicazione Windows contenuta in un solo eseguibile che non richiede alcuna installazione. Il file .ZIP che contiene la "distribuzione", contiene solo cinque file: l'eseguibile, il file di help in formato winhelp, una quick guide in formato txt e due file di esempio che possono essere caricati come immagine di memoria.

La scelta dell'autore è definire

una sorta di protocollo di salvataggio dei dati che prevede praticamente il listing esadecimale delle istruzioni.

Non è previsto il file ROM, che evidentemente è cablato nel codice. La ROM del KIM-1 prevede la presenza di un monitor di sistema per un totale di due Kbyte.

La versione del software è molto "basica", si tratta della 0.1a che evidentemente non ha avuto bisogno di grandi interventi dopo il primo rilascio ed è perfettamente utilizzabile e sufficientemente completa da permettere una efficacie presa di contatto con uno dei primi sistemi di sviluppo per microporcessori alla portata dei primi appassionati (si era nel 1976).

Il software ha una pagina Web di presentazione all'indirizzo http://www.crbond.com/soft6502.htm

Giustamente l'autore parla di simulatore piuttosto che di emulatore; infatti l'ambiente è molto diverso dall'hardware originale che era spartano all'eccesso. Qui ad esempio vengono visualizzati i contenuti dei registri, il registro di flag (con il led acceso nel caso di bit a 1), è disponibile una finestra scorrevole che visualizza la memoria a blocchi di otto byte ed infine le due porte principali del KIM evidenziate con altrettanti led rossi e verdi.

Al momento del lancio il monitor prende il comando (come dopo ogni reset (tasto RS) e restituisce una memoria pulita (tutto a 00) e il display si pone in modalità indirizzo con il valore sul display e il contenuto sui due display a destra che appaiono separati da quelli dell'indirizzo (i quattro di sinistra).

L'utilizzo è pari a quello che si effettuerebbe sulla macchina reale: la modalità DA inserisce il valore nella cella all'indirizzo esadecimale visualizzato e il tasto "+" da l'ok per la variazione del contenuto e sposta l'address di un passo in avanti.

Decisamente più comodo, anche se meno "reale" utilizzare la funzione di import di un file in formato h6x (come viene chiamato) che altro

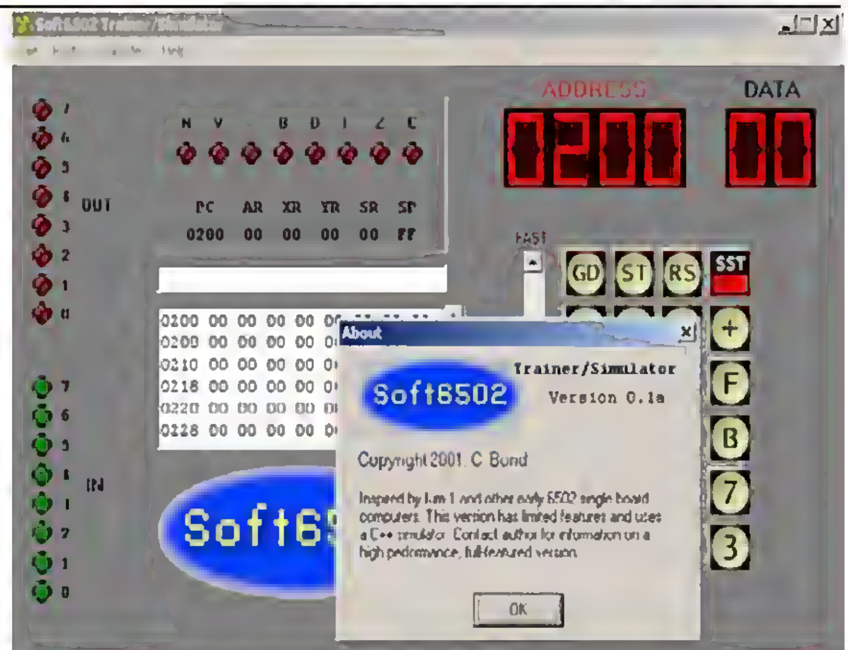


Figura 2. Il box informativo "About".

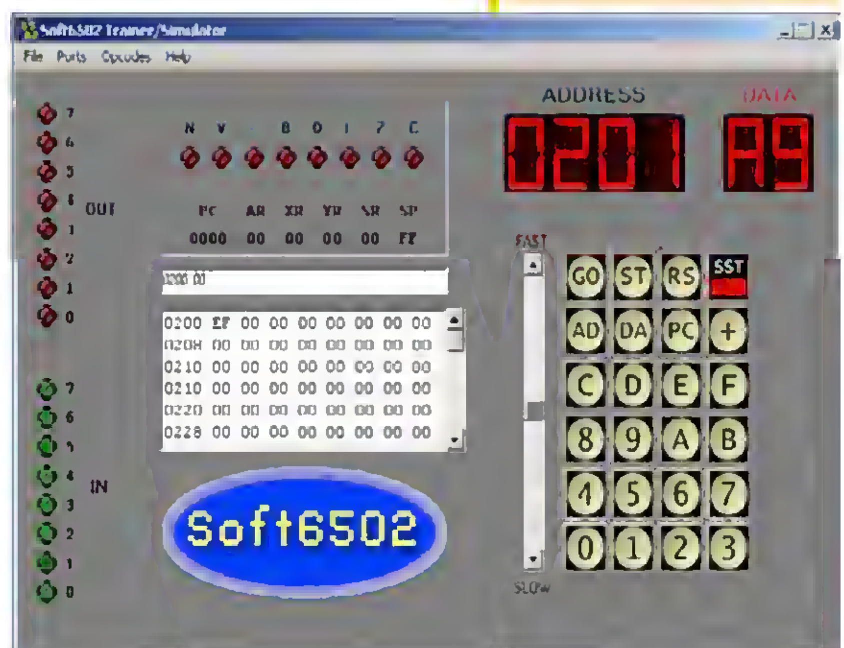


Figura 3. Stiamo inserendo i nostri primi dati.

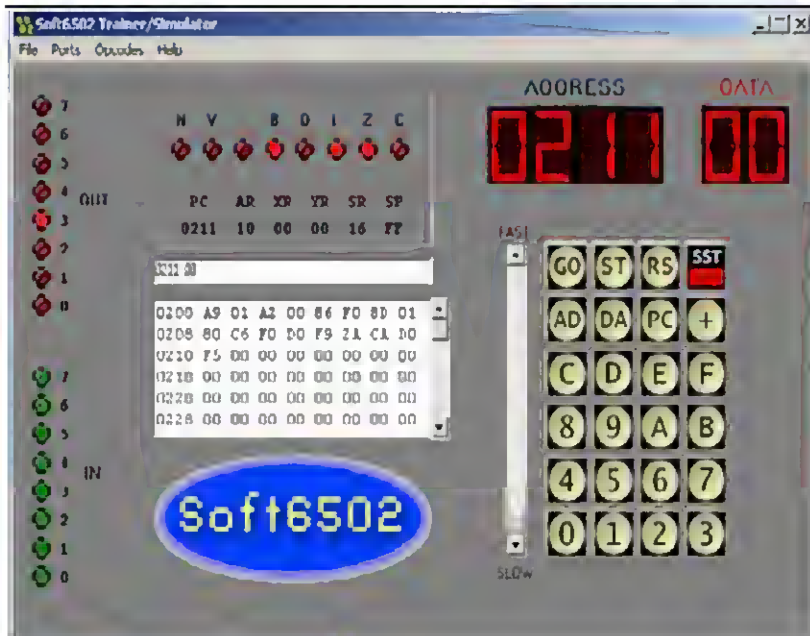


Figura 4.
Abbiamo caricato e lanciato il programma lights che looppa accendendo in sequenza i led della porta di output.

Figura 5.
L'attivazione della modalità Single Step viene evidenziata dalla spia verde sotto l'interruttore SST.

non è che una rappresentazione di dump della memoria come questa:

```
0200 A9 01
0202 A2 00
0204 86 F0
0206 8D 01 80
0209 C6 F0
020B D0 F9
020D 2A
020E CA
020F D0 F5
0211 00
```

Il dump è il listato del programma Lights65 fornito come esempio e che semplicemente accende in sequenza i led della porta di output in un loop infinito.

Per prendere confidenza con il simulatore ed iniziare a lavorare in codice macchina con il microprocessore, l'autore suggerisce un semplice programmino che semplicemente trasferisce in output il valore caricato nell'accumulatore.

La sequenza di operazioni per fare questo è (i tasti da premere sono in grassetto):

RS *(reset del sistema e inizializzazione all'indirizzo 0x0200 della memoria).*

DA *(tasto per abilitare l'inserimento dei dati);*

A9 *(codice della prima istruzione: caricamento immediato di un valore nell'accumulatore);*

+ *(tasto di conferma dell'avvenuta impostazione del valore);*

55 *(valore che sarà caricato nell'accumulatore);*

+

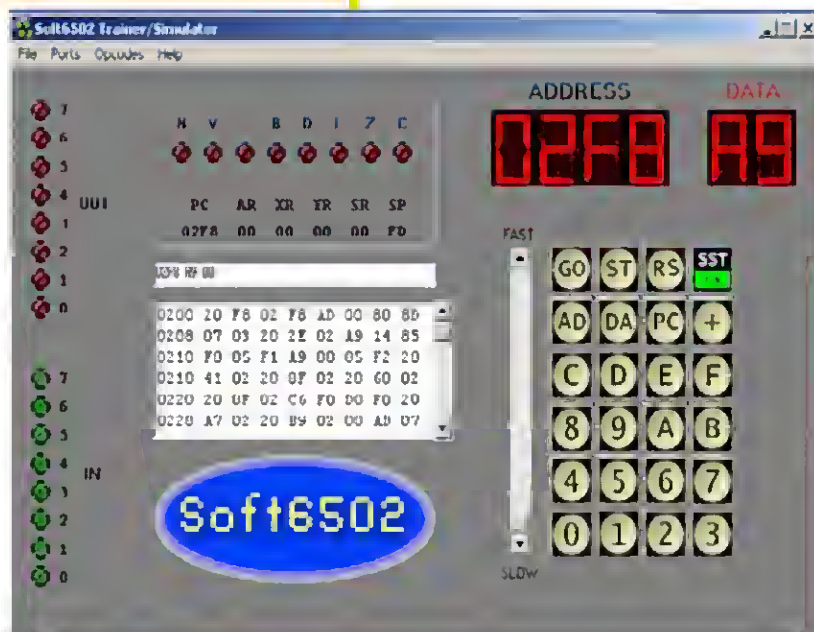
8D *(istruzione Store)*

+ 01 + 80 *(indirizzo di memoria dove inserire il valore);*

RS *(per riportare il Program Counter al valore di partenza 0x0200);*

GO *(per eseguire).*

Nella figura 6 il risultato dell'esecuzione con la porta in output che segnala il codice 55 (binario 01010101).



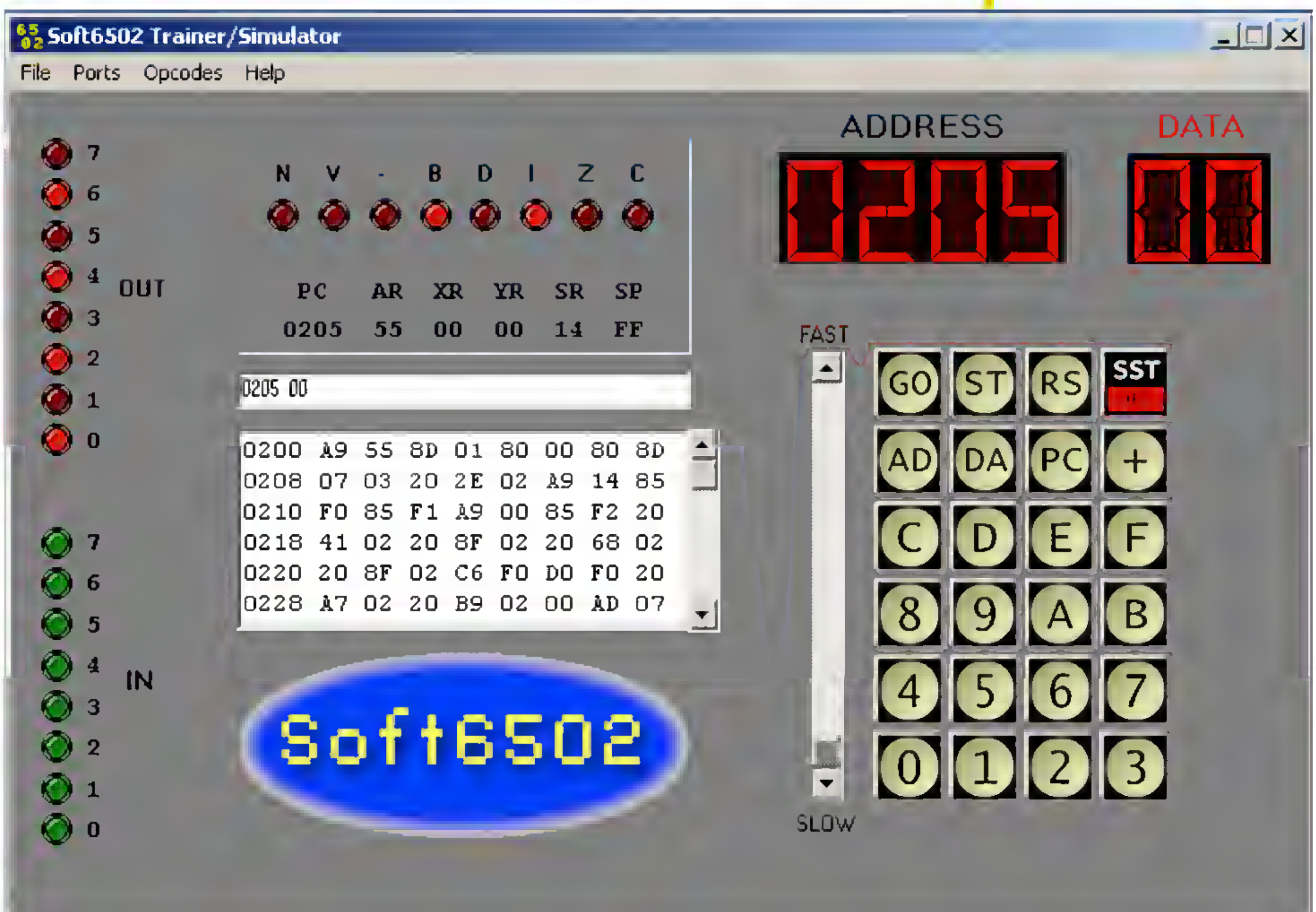
Conclusione

Un emulatore/simulatore che ha dell'essenziale, senza dubbio, ma che offre quello che promette: avvicinarsi alla programmazione del micro 6502 e assaporare (sebbene da lontano), l'atmosfera magica di una notte d'invernale quando fuori nevicava e vicino al caminetto accendevamo il nostro nuovo compagno di giochi: il KIM-1.

State bene.

[L2]

Figura 6. Il risultato dei nostri primi sforzi "programmatori": l'output de valore 0x55 sua porta di uscita.



Il racconto

Rubrica curata da [Bs]

Storie di vita dove i computer (soprattutto retro computer) c'entrano in qualche modo.

Automatik (2) - Game Star

Dove si racconta la mia esperienza di stagista alla Game Star.

Lo stage che doveva introdurmi nell'azienda che mi aveva scelto come "tecnico elettronico", pomposa definizione per dire "quello che avrebbe fatto di tutto compreso lo spazzare il magazzino", si svolse presso una ditta più o meno delle stesse dimensioni che aveva sede in Veneto, precisamente nel paese di Soave. Soave è famosa nel mondo perché vi si produce un vino bianco particolarmente gradevole e per chi frequentava il mondo dei noleggiatori di videogiochi, per essere sede della ditta che si chiamava qualcosa come "Game Star" o li vicino, diretta da un tizio che per essere buoni si potrebbe aggettivare come "trafficone". Egli si appellava Tiziano, ma ahinoi che distanza culturale dal suo nobile omonimo, sempre di origine veneta, autore di celebri dipinti!

Tiziano era uno di quei personaggi che non possiedono nulla, se non una gran massa crescente di debiti, eppure sembra che abbiano tutto: macchina prestigiosa con su una o due ragazze rigorosamente bionde e giovanissime, sempre vestito in giacca e cravatta con una noncuranza studiata, orologio d'oro e sguardo sorridente. Ne ho conosciuti due nella mia vita: Tiziano e un'altro tipo, un geometra, anch'egli perennemente in fuga dai creditori.

Questa "Game Star" era uno dei parti temporanei di colui che ospitò l'inizio della mia carriera, quanto sia durato il nome e la ditta è difficile dirlo ma sicuramente poco. Già allora ricordo che Andrea e Michele, i due dipendenti fissi, si preparavano a cercare un posto alternativo constatando la poca solidità dell'azienda dove lavoravano che si traduceva nel costante ritardo con il quale venivano regolati i loro salari. Anche la moglie e/o compagna (non l'ho mai capito) lavorava in ditta e precisamente in ufficio. Era una bionda "biondissima", di quelle che a incontrarle per strada non puoi fare a meno di seguirla con lo sguardo dopo che è passata. Questo non impediva al furbo Tiziano di eclissarsi con qualcuna delle sue amiche e starsene via magari tre giorni senza farsi vedere nel capannone. Non ho avuto modo di indagare quale fosse il menage famigliare della coppia, ma quello che notai fu che la moglie non non era affatto turbata dalle due "amiche" del marito e che sembrava non avessero altra occupazione nella vita che accompagnare Ti-

ziano sulla sua macchina sportiva. Quando Tiziano spariva per l'intera giornata o anche per più giorni, apparentemente nemmeno la signora sapeva nulla del marito/compagno, al punto che lo chiedeva a noi, semplici dipendenti, che poco aiuto potevamo darle. Chi apparentemente sapeva qualcosa più di noi e della moglie era Salvatore. Lo si capiva perché portava in laboratorio le televisioni che doveva riparare e non lo faceva mai quando Tiziano era in circolazione. Se Tiziano mancava la mattina e si vedeva venire Salvatore alle quattordici con un TV a colori sul carrello, si era certi che il capo non si sarebbe fatto vedere per tutta la giornata.

Per noi era una pacchia: semplicemente si passava il tempo a sfidarci a flipper o a qualche gioco nuovo appena montato. Ricordo ad esempio un gioco che aveva nome "Moon Ranger" (mi pare) che era una macchinetta che correva sulla superficie scabrosa di qualche satellite, la Luna immagino, sparacchiando a certe meduse che facevano la parte degli alieni e saltando per evitare le mine che i "cattivoni" avevano seminato sul percorso.

Per la verità ho scoperto solo ora, mentre cercavo delle immagini opportune per decorare il testo, che in realtà ce n'erano due di giochi, praticamente indistinguibili: il secondo si chiamava "Moon Patrol", sinceramente non lo ricordavo affatto questo particolare.

Quando iniziai la mia avventura in questo mondo, esattamente nell'autunno del 1982, questo Moon Ranger era lo stato dell'arte e la Game Star ne stava assemblando parecchi da rivendere poi ai vari noleggiatori che confluivano dal bacino del Triveneto-Emilia-Bresciano.



Imparai cos'erano i videogiochi in quelle tre settimane, prima li vedevo nei Bar senza dubbio, ma a parte il flipper e il calcetto non mi attiravano per nulla, anzi li trovavo poco divertenti e buoni per chi volesse buttare i propri soldi. E il flipper? Voi direte, il flipper non è una macchinetta mangiasoldi? E' vero, ma in qualche modo trovavo che l'interazione fisica con l'oggetto, il fatto che sia possibile al limite mantenere la pallina in gioco in maniera indefinita (quelli bravi), permette un approccio a "rischio calcolato". Poi è vero che la casualità è parte integrante del gioco del



flipper, così come il generatore di numeri casuali è componente aleatorio nei giochi arcade, ma volete mettere l'aspetto "mitico" cui è legata l'immagine del flipper? C'erano già allora centinaia di film che ne celebravano l'aspetto simbolico per non contare la serie Happy Days con Fonzie e compagnia.

Il sig. Tiziano era ai miei occhi il prototipo dell'imprenditore: una persona incapace di lavorare che sfruttava il lavoro altrui. Nel mio giovanile slancio idealista ero all'epoca molto vicino alle idee socialiste se non proprio comuniste, in seguito mi sono spostato di molto da questa parte politica e sono giunto sulla riva degli ignavi che non credono a nessuno e non trovano un rappresentante politico che sia pur lontanamente "decente" rispetto alle idee che vorrei condivise. Ma questo è un'altro discorso.

In seguito ho conosciuto anche imprenditori, soprattutto piccoli e artigiani pure, che riscattarono in me questo stereotipo del datore di lavoro. Ho conosciuto anche persone che lavorano in proprio ma che sono onesti con i dipendenti e con lo stato, insomma che sono un bene per la società.

Tiziano aveva il dono della simpatia e anche con me fu sempre gentile e corretto. Si avvicinava a volte mentre ero intento a trafficare in qualche cabinet per fissare una scheda o avvitare un pulsante e mi chiedeva come stesse andando. Io rispondevo che andava benissimo, che mi piaceva e che speravo di poter continuare a fare quel lavoro. Lui se ne andava soddisfatto non prima di avermi gentilmente fatto notare che avrei dovuto farne della strada per "essere utile all'azienda"; era una delle sue frasi preferite.

L'utilità all'azienda era nella sua idea il motore dell'economia. Erano utili all'azienda chi veniva a lavorare senza fare assenze per malattia, quelli che si fermavano fuori orario senza chiedere straordinari e quelli che si sobbarcavano quattordici ore di carico e scarico di materiale pesante come i pezzi dei biliardi o certi cabinet di videogiochi massicci oltre misura che sembravano armadi. Ricordo a questo proposito un cabinet doppio con due volanti e altrettante pedaliere che ospitava due monitor con un gioco di corsa di automobili. La grafica faceva ridere già allora: semplici tracce del circuito e cinque quadratini per le macchine (quattro per le ruote e uno per il corpo macchina). Però ci si giocava in due con un solo gettone e si facevano sfide vere e proprie. L'unico lato negativo di questo intrattenimento era riservato a noi che dovevamo spostarlo in giro per i bar e sale giochi: in pratica era come un pianoforte verticale, di quelli però full, alto circa due metri e largo uno e mezzo per circa ottanta centimetri di profondità. Non ricordo come si chiamasse, qualcosa come "car racing" mi sembra di ricordare...

Il mio futuro datore di lavoro passò un pomeriggio nell'azienda di Tiziano per vedere come me la cavavo e per il fatto che trafficavano assieme nel settore ed avevano quindi sempre macchine da scambiare e/o comprare e vendere. Stettero parecchio in ufficio lui e Tiziano mentre io facevo finta di non accorgermi che ogni tanto lanciavano degli sguardi nei miei confronti. Intanto io mi applicavo con più lena del solito attorno alle macchine fingendo una sicurezza e una capacità lavorativa che all'epoca era solo simulata. Particolarmente d'aiuto erano per me i test automatici nei flipper elettronici. Ogni flipper conteneva in firmware una sequenza di test attivabile con un pulsantino interno e che serviva alla scansione ciclica degli switch nel sistema. In pratica questo test accendeva in sequenza le lampadine e faceva scattare le bobine in maniera tale che l'operatore potesse rendersi conto del perfetto funzionamento del tutto. Bisogna pensare che su un flipper c'erano circa 100 lampade (ora sono led) e almeno una

ventina di bobine. Poiché il test veniva fatto in sequenza con un certo tempo di delay fra una accensione e la successiva, lanciarlo equivaleva ad avere dai cinque ai dieci minuti buoni per stare a guardare dandosi un contegno, pur senza fare nulla.

Alla fine il mio prossimo datore di lavoro uscì dall'ufficio del collega-concorrente e mi annunciò che se volevo potevo prendere servizio appena possibile. Io ringraziai e mi sembrò in quel momento di essere arrivato; era una soddisfazione intima perché ero riuscito a fare, o almeno lo credevo allora, quel famoso salto di qualità nella vita che apre la strada al cambiamento.

Ora, pensai, dipende solo da te, datti da fare. Dovrai studiare, applicarti anche oltre l'orario di lavoro, spremerti le meningi per trovare idee e soluzioni. In una parola farti una strada.

Licenziandomi e parlando con i prossimi ex colleghi di lavoro mi veniva naturale darmi delle arie, che già allora sapevo esagerate e anzi sospettavo nascondessero qualche magagna, come poi puntualmente si verificò.

Dovevo passare due o tre settimane di preavviso, non ricordo di preciso quante e fu questo periodo che mi portò i primi dubbi e le prime paure.

Come recita il noto proverbio "Chi lascia la via vecchia per la nuova, sa quel che lascia ma non sa quel che trova", mi dibattevo nel dubbio fino a perderci il sonno. E' vero che era quello che avevo voluto, ma questo futuro, o spiraglio del futuro, si presentava meno chiaro di quanto avrei bramato. Nelle ore passate ad assemblare schede e programmare lo Z80 al punto che ne conoscevo il codice macchina a memoria, avevo pregato che questo fantastico hobby potesse avere per me uno sbocco professionale. Ora però, da quel che avevo visto alla Game Star, non era tutto oro quel che luccicava e c'era da mettere in conto il pendolarismo con perdita di tempo e costi di automobile.

Improvvisamente tutti i difetti del vecchio lavoro che stavo lasciando non mi sembravano così terribili e ne mettevo vicini i pregi: il tempo libero, il non dover spostarmi, mangiare fuori... Ma ormai il dado era tratto, ero in ballo e dovevo ballare,... e via dicendo ne potrei recitare decine di aforismi e motti che la saggezza popolare ha raffinato in millenni di cultura.

Passai la Domenica precedente il giorno in cui dovevo presentarmi sul nuovo lavoro a meditare. Passeggiai a lungo per le strade semideserte del paese e finì su sul monte, la dove avevo eletto a rifugio personale un angolino poco noto dal quale si poteva ammirare l'intera valle.

Laggiù, sul fondo, aldilà del monte che ospitava su una sua sella il passo che tutti i giorni avrei poi percorso, c'era la città e il nuovo lavoro con nuovi colleghi, sfide, successi (forse) e sconfitte (certe). Sentivo la svolta che si approcciava alla mia vita e sapevo che non sarei più potuto tornare indietro.

[Lp]

Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Storia dell'interfaccia utente

Introduzione

Lo sviluppo tecnologico in qualsiasi campo applicativo è sempre andato di pari passo con una evoluzione degli strumenti utili per controllarlo al meglio. Quando vediamo le fotografie delle prime automobili ci sempra incredibile che per guidarle si fosse pensato ad una manovella-manubrio piuttosto che al comodissimo volante! Eppure è così, sono passati anni prima che a qualcuno venisse questa brillante idea, mai più abbandonata, segno tangibile che è difficile fare di meglio.

Nell'ambito informatico è accaduto più o meno la stessa cosa: una ricerca continua del miglior modo per comandare questa nuova macchina dalle innumerevoli possibilità.

All'inizio si è usato quello finora disponibile in altri comparti simili, come la telescrivente o ancora prima dei semplici bottoni e interruttori, poi un miglioramento dei mezzi fisici di interazione oltre alle modifiche di utilizzo degli stessi in una continua rincorsa a convincere l'utente ad adottare le soluzioni via via proposte.

Questa del coinvolgimento dell'utente è essenziale: la migliore interfaccia possibile potrebbe non essere quella preferita dai più. Pensiamo ad esempio ai comandi vocali: sembrerebbe non esistere metodo migliore! Nessun tasto da pigiare, velocità di scrittura inarrivabile sulla tastiera, etc... eppure non è che questo metodo del controllo del PC sia poi così diffuso, anzi!

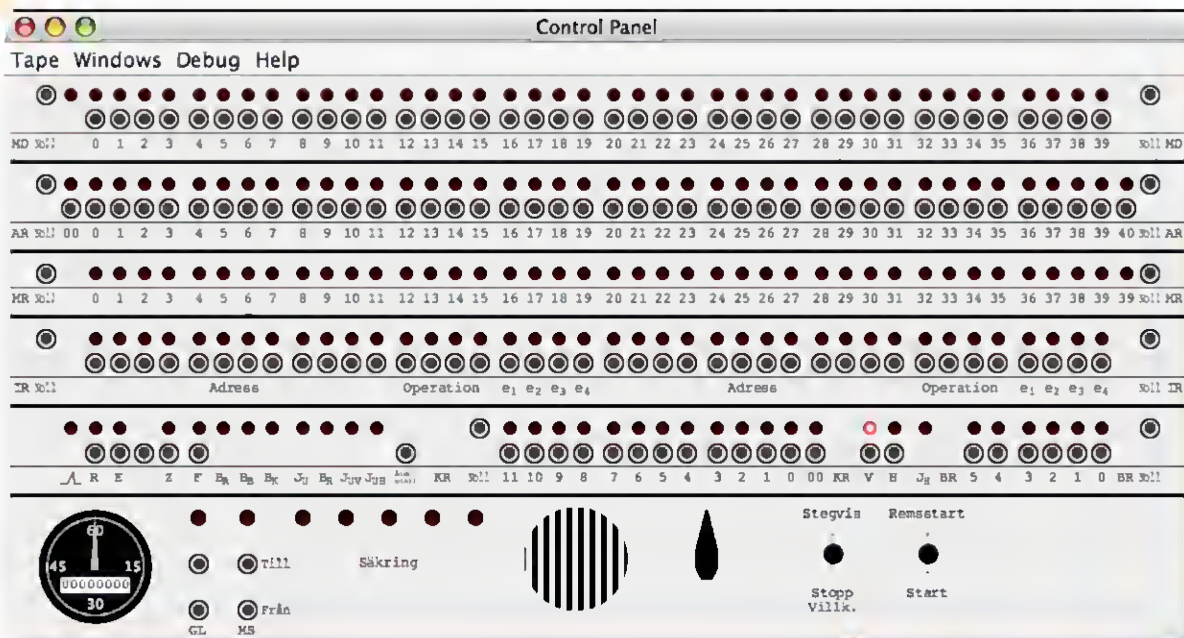
Quello che vogliamo realizzare in questo articolo è un viaggio nell'evoluzione delle interfacce; una breve rassegna di come eravamo e di come siamo poi diventati, con uno sguardo ai possibili futuri.

Prima del terminale

Prima del terminale con tastiera erano solo interruttori e lampadine.

Quando le possibilità di calcolo erano limitate e il ciclo di utilizzo semplice al massimo, quello che importava era un sistema a basso costo che permettesse di impostare dei valori elettrici (l'input) e leggere altri valori elettrici (output).

Quello che caratterizza questo metodo di controllo è la bassa nu-



merosità: se le informazioni da impostare e da leggere sono minime, si può anche pensare che non sia poi così tedioso agire su switch e pulsanti ed infine ottenere la risposta della macchina da semplici lampadine, al limite colorate.

Un esempio è l'interfaccia del calcolatore SMILL (JN ne ha ospitata la recensione di un emulatore), visibile nell'immagine a fianco.

Siamo già in presenza di una certa complessità di interpretazione, dal momento che i valori di tensione sono impostabili bit per bit all'interno dei registri. Questa è una macchina di tipo "Von Neuman", quindi già un calcolatore "moderno" a tutti gli effetti.

Il pannello di controllo del calcolatore SMILL ci permette di fare una considerazione sul livello "software" dell'interfaccia. Qui abbiamo una esposizione a livello di bit del contenuto dei registri, quindi all'operatore è richiesto di lavorare molto vicino al livello macchina.

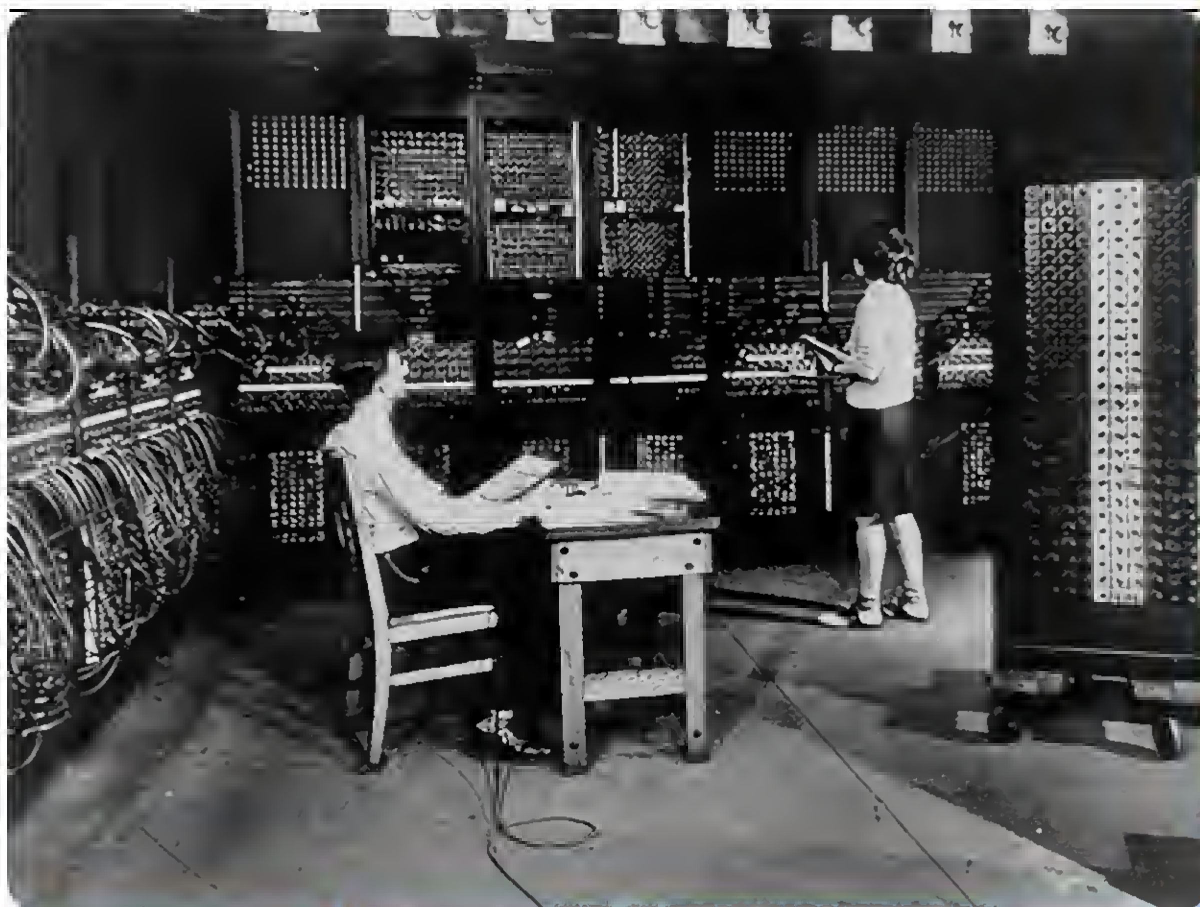
Il pannello di controllo espone due registri a 32 bit (prime due righe in alto) con luce accesa uguale bit a valore logico uno e spenta per il valore zero. Il pulsante accende/spegne il corrispondente bit.

La terza riga, partendo dall'alto, è il registro "risultato", solo in output (mancano i pulsanti). Segue il pannello del programma con indirizzo ed istruzioni ed infine un pannello di controllo per impostare i parametri di esecuzione, far partire e interrompere il programma, etc…

Siamo già "avanti" nell'interazione uomo-computer e si nota già in questa interfaccia il modello che sarà perfezionato più avanti: l'utente controlla dei valori in registri della macchina, imposta le istruzioni in corrispondenza degli indirizzi di memoria-programma e agisce con dei comandi che ordinano l'esecuzione delle istruzioni.

Nell'interfaccia SMILL c'è anche un'altra proto-idea di interazione: agire a livello globale di registro piuttosto che a livello di bit. Nello SMILL questo è limitato all'azzeramento del contenuto, realizzato dal pulsante "solitario" che sta all'estre-

La spartana interfaccia utente del computer SMILL.

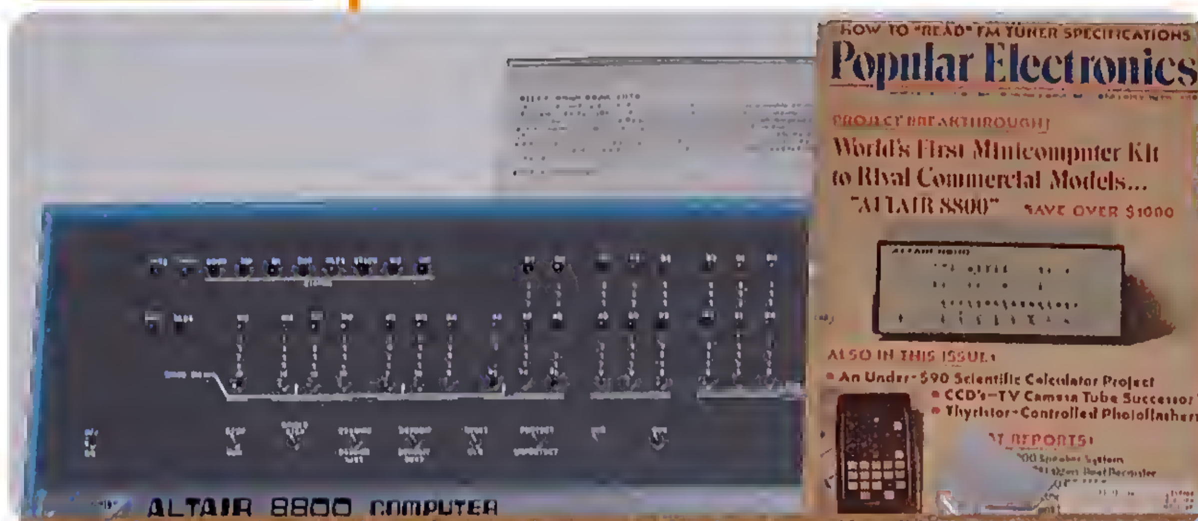


La sala macchine di un mainframe di prima generazione

ma sinistra di ogni pannello.

Ovvio che l'interfaccia SMILL è solo un esempio delle soluzioni comuni negli anni '50 e '60, precedentemente l'interazione uomo-macchina ha conosciuto un altro periodo significativo che si caratterizza dalla presenza di un mediatore (chiamato "operatore") il quale agisce su interruttori, pulsanti e qualche volta perfino collegamenti, seguendo le istruzioni che gli vengono impartite da un "team" di programmatori. Chi non ha mai visto le classiche foto dei sistemi ENIAC anni cinquanta? Quello che viene mostrato frequentemente è il lavoro umano, spesso fatto da figure femminili, nei confronti della macchina.

Anche i primi personal, spesso poco più che dei kit, richiedevano una qualche interazione manuale per impostare i dati di boot. Tipico esempio è l'Altair 8800 dove si impostano gli indirizzi di esecuzione del programma e poi si lancia e dove l'uscita su terminale video (tipicamente un TV domestico in bianco/nero) è una espansione del progetto base.



L'Altair 8800, uno dei primi kit di autocostruzione

La carta



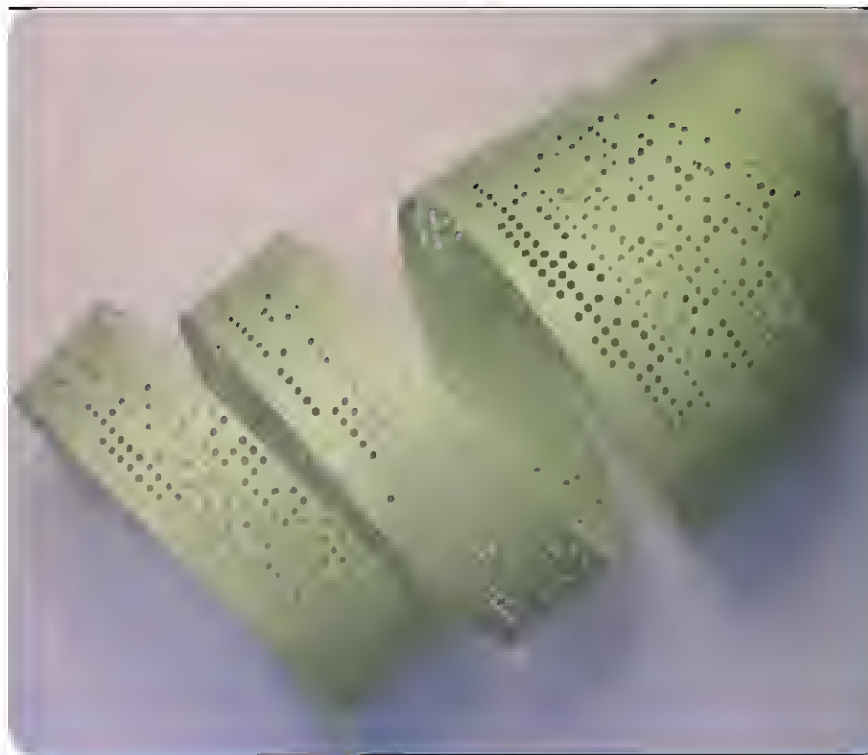
Una macchina perforatrice di schede

Il secondo passo verso una interazione uomo-macchina più produttiva delle semplici manovre meccaniche, si è avuta con l'adozione di stampati in output e di sistemi elettro-meccanici per quanto riguarda l'input dei dati.

L'output su carta è stata una evoluzione abbastanza lineare: in fondo esistevano le macchine da scrivere e le tecnologie di stampa più o meno meccanizzate come quelle in uso nelle tipografie e nei giornali. Per quanto riguarda invece l'input ci si è affidati ancora una volta alla carta: schede perforate e nastro perforato sono i due sistemi più noti. Che ci crediate o no, io stesso ho lavorato nei primi anni di università sulle perforatrici di schede. Si era attorno al 1974-75 e quelle disponibili (ricordo che ne avevamo due in facoltà, sempre occupate peraltro) erano già "moderne" per così dire. Infatti ogni scheda veniva richiamata automaticamente in posizione di perforazione e ordinata infine nel cassetto di output da dove si trasportava al lettore vero e proprio attaccato al mainframe.

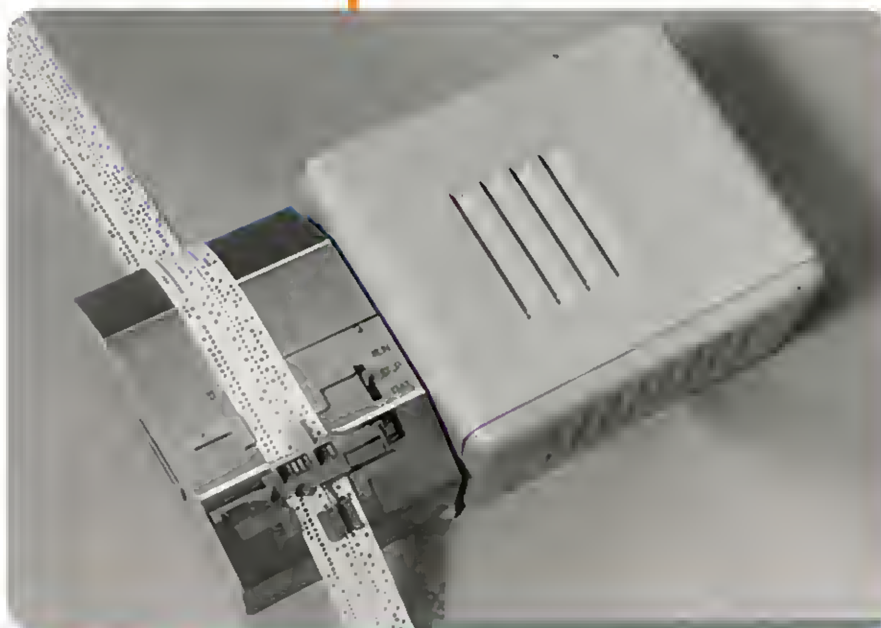
L'input con scheda perforata possiamo dire che era un sistema misto: meccanico-umano. Infatti si passava un sacco di tempo a riordinare le schede (guai se ti cadeva il mazzo di schede per terra! Ci si poteva impiegare ore a rimetterle in sequenza). Classica poi l'adozione di protocolli visuali che aiutavano gli operatori, dal momento che il lettore di schede ne poteva contenere centinaia, anche di programma diversi. Da noi all'università ricordo che la prima scheda doveva essere di colore verde e l'ultima rossa.

Il nastro di carta perforato ha risolto il problema dell'ordine delle istruzioni ma ovviamente ne ha messo in luce altri. Il più fastidioso era l'incidente che provocava lo strappo del nastro, nel cui caso era necessario procedere ad una lenta



Il nastro perforato conserva tutt'oggi un certo fascino. Sotto: una testa di lettura per nastro perforato.

azione di restauro muniti di forbici e nastro adesivo. L'operazione non sempre riusciva o era possibile e di conseguenza bisognava ridigitare tutto il programma da capo o, se possibile, farlo stampare dal sistema. Il nastro aveva anche un altro inconveniente: teoricamente si poteva piegare, ma in realtà era meglio conservarlo su bobina. Quindi un problema di ingombro oltre che di caricamento del lettore, meno agevole del pacco di schede perforate.



Schede perforate e nastro di carta hanno avuto una evoluzione parallela quando è entrata la tecnologia magnetica. Allora al posto dei buchi le schede portavano una banda magnetica (un po' quello che succede sulle carte del bancomat) e il nastro di carta era diventato una bobina di nastro magnetico.

Il nastro magnetico è una tecnologia adottata subito nei sistemi commerciali come i grandi mainframe o i mini dipartimentali, però con l'avvento degli home a basso costo la tecnologia a nastro comune nei sistemi di classe aziendale, era decisamente improponibile. Ricordo una unità a nastro da un quarto di pollice per il mini Olivetti M60 (anni 1985-90) che costava una ventina di milioni di lire (l'unità centrale ne costava 50).

I sette segmenti

Attorno agli anni '60 si erano rese disponibili per le costruzioni elettroniche le Nixie. Queste erano in pratica dei tubi a vuoto entro i quali trovavano posto dei filamenti sagomati in modo da visualizzare le cifre decimali e che venivano accesi in base al segnale applicato ai piedini della valvola stessa.

Esse erano usate negli strumenti di misura ma sembravano perfette come display per il calcolatore. I tecnici informatici stavano

cercando qualcosa che sostituisse la stampante della telescrivente (teletype) allora in uso. Essa era ingombrante e cosa ancora più importante, molto rumorosa. Avere la possibilità di mostrare un output su un display, per quanto rudimentale, che si aggiornasse in maniera dinamica, era proprio quello che ci voleva!

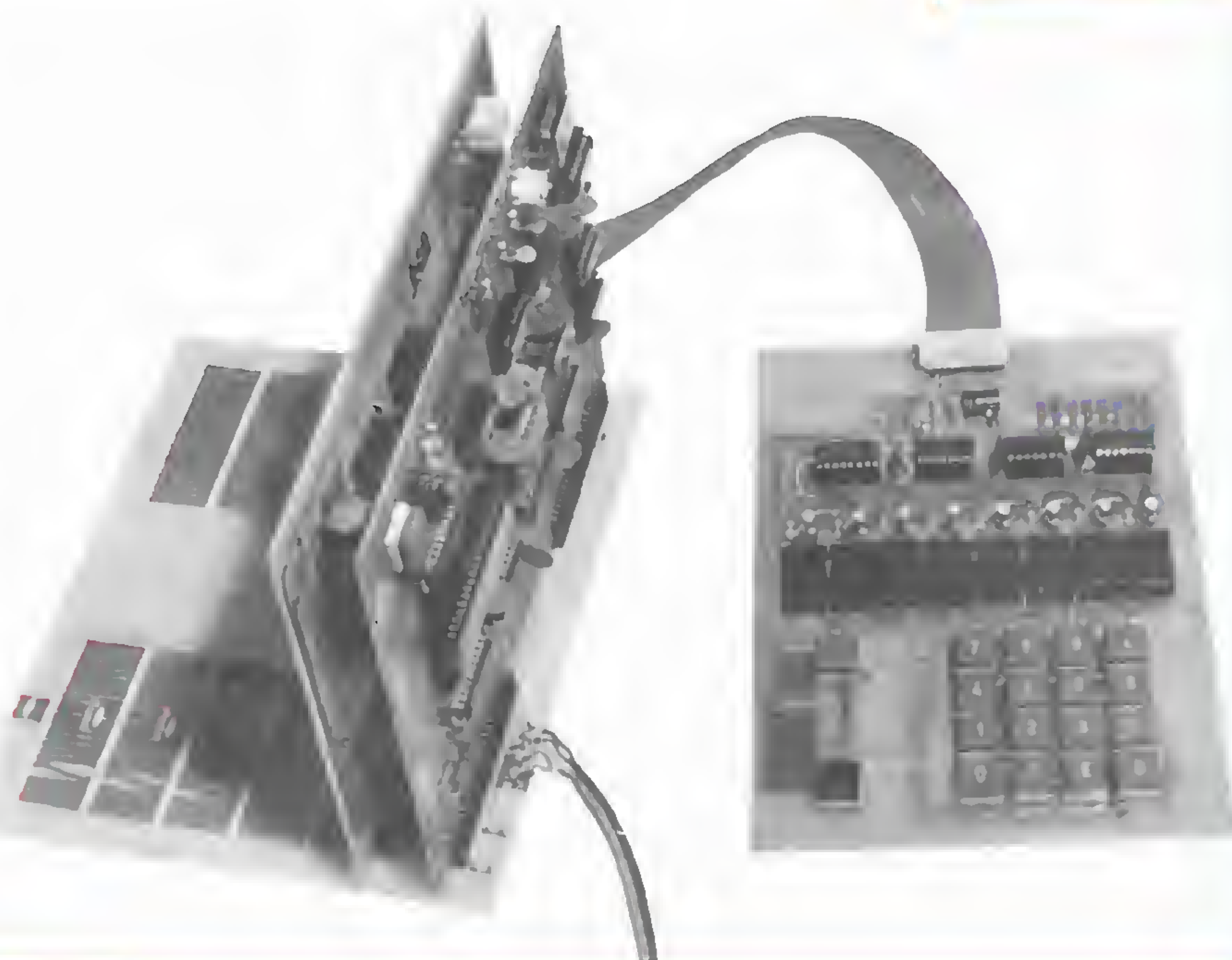
Furono così progettate delle nixie contenenti anche lettere dell'alfabeto ma era impossibile arrivare ad un vero display alfanumerico completo. La dimensione della valvola sarebbe stata troppo grande e anche la visibilità del filamento posto dietro tutti gli altri che rimanevano spenti, era compromessa se c'erano troppi strati.

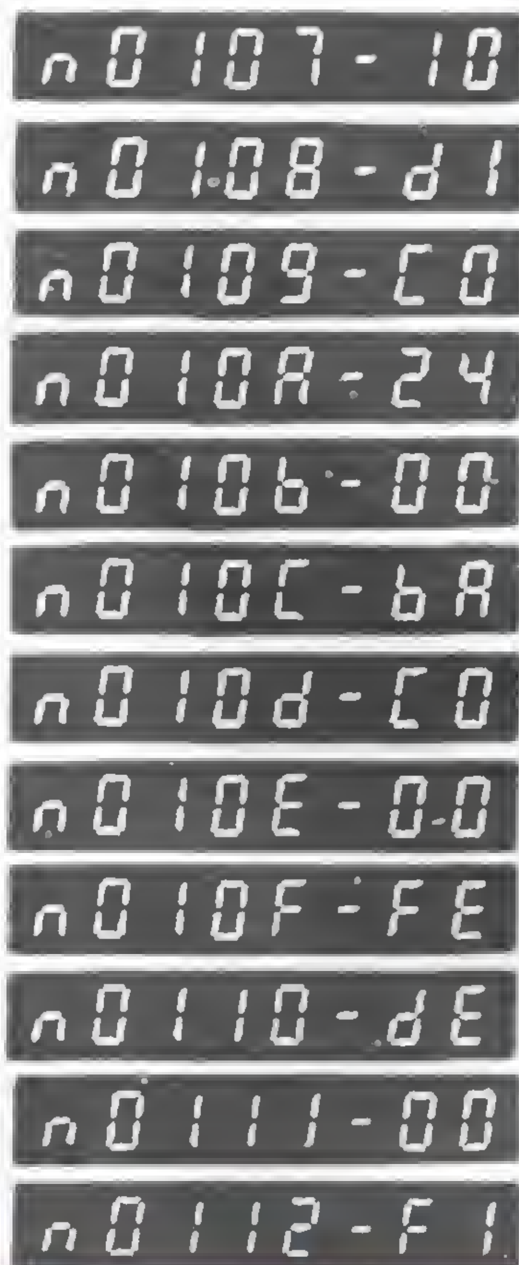
Il mondo era pronto per un'altra scoperta: il display a sette segmenti. Sembra banale, ma sostituire una unica accensione di un filamento all'interno di una nixie con una logica che attivasse solo i segmenti giusti, non deve essere stato un passaggio proprio banalissimo!

Il display a sette segmenti ebbe grande fortuna per due ragioni principali: per la disponibilità della tecnologia led che ne permise la costruzione contenendo dimensioni e consumi e la possibilità, seppure rozza, di rappresentare anche alcune lettere dell'alfabeto, oltre ad una serie di simboli più o meno grafici.

Era quello che l'informatica sta-

Il kit NEZ80 usa come input/output tastiera esadecimale e display a sette segmenti





Cifre e lettere ottenute giocando sull'illuminazione dei segmenti

va aspettando per abbandonare definitivamente la notazione ottale con la più efficiente esadecimale.

Spieghiamo per chi non conosce i dettagli della faccenda.

Tutti sanno naturalmente che il calcolatore funziona in logica binaria, questo da sempre, sebbene siano esistiti tentativi di introdurre una logica che permettesse l'utilizzo di più stati, ad esempio livelli di tensione, in modo da trasferire la notazione decimale direttamente nelle macchine.

Ora accomodando i bit in sequenza, ad esempio 4, si ottengono 16 combinazioni di possibili stati 0-1 e quindi sono rappresentabili sedici diversi simboli. Si è quindi pensato che era conveniente adottare una notazione aritmetica che prevedesse sedici simboli e non già i dieci cui siamo abituati dalla numerazione araba. I simboli partono con lo zero, proseguono con le cifre fino al nove per andare ancora avanti con A, B, C, D, E e F.

Con 8 bit a disposizione nella "parola" (così viene definito il parallelismo all'interno del microprocessore), si raggiungono le 256 combinazioni (da 0000 0000 a 1111 1111).

Usare sedici cifre al posto delle consuete dieci sembra sia un esercizio piuttosto difficile: in fondo noi abbiamo dieci dita! In realtà ci si abitua abbastanza velocemente ed è per questo che la notazione esadecimale (cioè in base sedici) è stata universalmente adottata.

Nei kit "prima maniera" era obbligatoria la tastiera esadecimale quale periferica di input, tastierino che faceva solitamente pandan con una striscia di display a sette segmenti.

La semplicità del tutto, ma anche la funzionalità, è apprezzabile dall'immagine "dal vivo" del kit Z80NE di Nuova Elettronica, mostrata a grandezza naturale nella pagina a fianco.

(continua...)

[Tn]



La scheda "tastiera esadecimale" del Kit di Nuova Elettronica

DIR

Le pagine dedicate al sistema operativo CP/M

Uno sguardo all'MP/M

L'MP/M è la sigla che significa "Multi-Programming Monitor Control Program" e che è stata attribuita dalla Digital Research nel 1979 al progetto di estensione del sistema operativo CP/M.

Dopo gli indubbi successi del CP/M, adottato in qualche maniera (nativa o adattata) praticamente da tutti gli home e personal computer che disponevano di una CPU compatibile, già alla fine del '79 si approcciavano sul mercato le richieste di disporre di un sistema operativo in time-sharing adatto ai sistemi con microprocessore.

Da una parte c'era chi era convinto che la CPU come lo Z80, nelle sue release più avanzate (A e B rispettivamente a 4 e 6 MHz), avrebbe sopportato ben altri carichi rispetto alle limitate incombenze derivanti dalla gestione della monoutenza; dall'altra c'era chi voleva comunque proporre un sistema "allargato" con terminali e tutto il resto per attaccare il mercato dei piccoli uffici ai quali magari il personal andava stretto e il mini dipartimentale troppo largo.

L'idea della DR (Digital Research) fu quello di non discostarsi molto dall'impianto del CP/M originale, magari conservando la compatibilità del nuovo sistema con il suo predecessore.

Questo anelito di compatibilità verso il passato è un obiettivo che i progettisti hanno sentito come punto di forza del sistema che andavano progettando e che ha "salvato" l'MP/M dal venire travolto da altri sistemi operativi ben più carozzati come quelli derivanti da UNIX (Xenix in prima linea).

Un sistema in time-sharing non è facile da progettare e realizzare. Prima di tutto bisogna confrontarsi con i concetti di divisione delle risorse da assegnare ai vari task e di concorrenza rispetto alle richieste di accesso alle periferiche.

La complessità che hanno incontrato i progettiti nel passare dal monoutente CP/M al "fratellino maggiore" MP/M è ben evidenziata negli schemi funzionali visibili nella pagina seguente. La scelta dei progettisti è stata quella di conservare l'impianto modulare di base con i tre moduli specializzati (SHELL, BIOS e DOS) complicandoli il necessario per tenere conto che sul sistema operano in contemporanea task diversi e utenti in concorrenza.

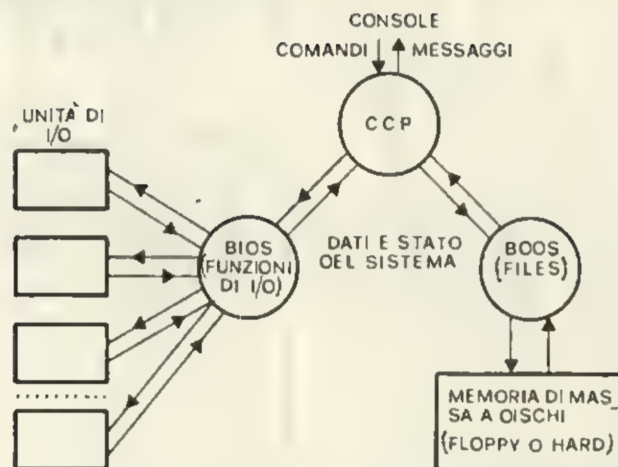
Quando si introduce in un siste-

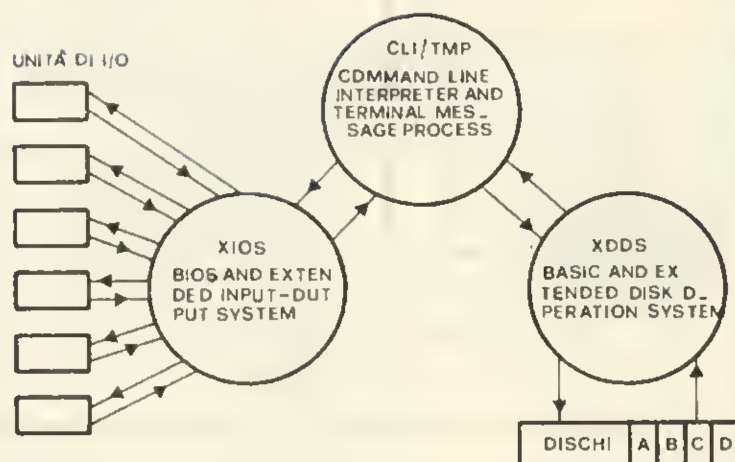
ma il concetto di concorrenza, cioè di disputa all'uso di una risorsa da parte di più richiedenti, ci si possono inventare mille modi per portare avanti una strategia che consenta l'ottimizzazione di tale risorsa. Il metodo più semplice è fare delle semplici code (una per ogni risorsa) nel quale sono registrate le richieste di accesso di ciascun task nell'ordine con il quale arrivano. E' una pila di tipo FIFO (First In First Out), il primo che arriva è il primo che esce, come il fluido in un tubo.

Questo semplice metodo ha delle inefficienze tali che in pratica non è possibile adottarlo e devono essere messi in campo delle modifiche che consentano una gestione più flessibile della coda di richieste. L'MP/M adotta il concetto di priorità per consentire ad un task di "scavalcare" altri task che hanno meno urgenza.

Senza entrare in ulteriori particolari quello che ci premeva far capire è il compromesso adottato dai progettisti che si sono trovati difronte a questi problemi. Nell'MP/M i task ad esempio sono al massimo otto e le priorità vanno dal livello 0 (la più elevata) al livello 4 (la meno urgente). Oltre a questi concetti elementari è necessario gestire lo stato dei singoli processi per consentire ad esempio la messa in sleep di certi lavori che sono in attesa di altre elaborazioni prima di continuare.

Esiste anche nell'MP/M ovviamente il cosidetto task-descriptor, cioè la struttura dati conservata in memoria che riporta i dati di elaborazione del task. Il sistema progettato dall DR prevede di assegnare porzioni fisse di memoria al singolo task ed esattamente 48 Kb per task. E' una scelta che oggi sarebbe inaccettabile ma che semplifica molto la gestione della memoria centrale da parte del sistema operativo. Prima di tutto è possibile impostare dei semplici switch di banco per passare da una porzione all'altra della memoria e lascia una porzione di 16 Kb di memoria comune a tutti i banchi dove troveranno posto i dati e i task riservati





I due schemi, presi da una pubblicazione dell'epoca, evidenziano la differente implementazione dei moduli base del sistema operativo nel caso del CP/M (in alto) e dell'MP/M (in basso).

al sistema stesso.

Un'altro problema è la gestione delle memorie di massa, cioè i dischi floppy e hard-disk. I progettisti hanno deciso di conservare una compatibilità massima con l'esistente CP/M e pertanto i descrittori dei file su disco e il metodo di registrazione è compatibile nell'MP/M con il predecessore.

Il CP/M aveva già in se la possibilità di dividere un disco in "Aree utente" (User Area) nominate da 0 a 15. In pratica era possibile creare sul disco delle zone riservate ai vari utenti e passare da una all'altra con comandi da console. Se per il CP/M la cosa era opzionale e non esisteva nessun meccanismo di protezione fra le are, nell'MP/M questo concetto viene personalizzato e l'utente poniamo numero 7, accederà all'area disco ad esso associata UA7 (User Area 7).

In realtà di default la separazione delle aree disco fra utenti diversi non è rigida: un utente può accedere anche alle aree degli altri.

Sul sistema "gira" sempre un utente 0 e un task 0 che serve come controllo o "console", come si diceva una volta.

I comandi che l'utente si trova a dover utilizzare sul suo terminale sono gli stessi che usava per il CP/M con l'aggiunta di quei comandi più sofisticati e riservati ad un operatore che manipolano i task, le priorità, etc...

Ecco la sintassi e la descrizione di alcuni comandi.

Il più semplice è sicuramente CONSOLE che semplicemente stampa il numero della console sulla quale si sta operando.

ATTACH: permette di attaccare e distaccare (DETACH) un processo dalla console.

USER visualizza l'area disco (e permette anche di cambiarla) alla quale si sta accedendo.

Per il sistemista (chiamiamolo così) il comando più utile è MPM-STAT che da una detagliata informazione sullo stato dei processi, le code, i semafori, etc...

L'MP/M è in realtà un super-set del CP/M e la riprova è che si genera da un CP/M funzionante sulla macchina (comando GENSYS) e viene lanciato dallo stesso CP/M dopo il suo caricamento (comando MPMLDR).

L'MP/M offre una serie di servizi aggiuntivi rispetto al predecessore; elenchiamo i principali:

- *Supporto multi-terminal*
- *Multi-programming su ogni terminale*
- *Supporto al bank switching della memoria*
- *protezione della memoria*
- *Gestione della concorrenza*

sui processi della CPU e sulle operazioni I/O

- *Interprocessing comunication*
- *Gestione del polling e della gestione degli interrupt*
- *Configurazione dinamica del sistema al caricamento.*

Al momento della partenza del sistema operativo, la Shell prende il controllo ed espone il Prompt in attesa di processare i comandi dell'utente. All'accensione il sistema si dispone sulla console numero 0 e il prompt sarà: "0A>" dove il numero 0 indica la User-area che viene usata di default sul disco A:.

Con il comando USER ci si può spostare su qualsiasi delle 16 console:

```
0A>USER 3
user = 3
3A>
```

Il sistema risponde stampando lo user di destinazione e poi nuovamente il prompt.

Più di un programma può essere lanciato da una delle 16 console (numerate da #0 a #15). Quando un programma è in esecuzione può essere "staccato" dalla console con la combinazione di tasti CTRL-D e successivamente "riattaccato" alla stessa console con il comando ATTACH seguito dal nome del programma. Quando un programma è stato "detaccato" continua a funzionare in modalità background, come su un normale sistema multitasking a divisione di tempo.

Nello screen in questa pagina abbiamo riportato l'output che si ottiene chiedendo la stampa dello stato del sistema (comando MPMSTAT), dove sono elencati i processi attivi e tutte le altre informazioni utili all'amministratore di sistema.

Conclusione.

Abbiamo dato un semplice sguardo alle caratteristiche principali del sistema operativo MP/M, ragionando soprattutto rispetto alle differenze con il predecessore CP/M.

Per ora ci fermiamo qui ma certo l'argomento merita ulteriori approfondimenti che ci ripromettiamo di analizzare in prossimi articoli.

```
****** MP/M Status Display ******

Top of memory = FFFFH
Number of consoles = 02
Debugger breakpoint restart # = 06
Stack is swapped on BDOS calls
Z80 complementary registers managed by dispatcher
Ready Process(es)
     MPMSTAT        Idle
Process(es) DQing:
     [Sched     ]   Sched
     [ATTACH    ]   ATTACH
     [CliQ      ]   cli
Process(es) NQing:
Delayed Process(es):
Polling Process (es)
     PIP
Process(es) Flag Waiting:
     01 - Tick
     02 - Clock
Flag(s) Set:
     03
Queue(s):
     MPMSTAT        Sched     CliQ ATTACH     MXParse
     MXList         [Tmp0     ]MXDisk
Process(es) Attached to Consoles:
     [0] - MPMSTAT
     [1] - PIP
Process(es) Waiting for Consoles:
     [0] - TMPO          DIR
     [1] - TMP1
Memory Allocation:
Base = 0000H        Size = 4000H   Allocated to PIP     [1]
Base = 4000H        Size = 2000H * Free *
Base = 6000H        Size = 1100H   Allocated to DIR     [0]
```

Retro Linguaggi

LISP (parte 1)

La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.



Il logo della International Lisp Conference svoltasi nel 2009.

Introduzione

E' incredibile: si pensa che un linguaggio di programmazione sia morto e sepolto e invece cercando un poco in Rete scopri che non solo la supposizione è falsa, ma anzi dovresti ritornare a programmarci per ottenere il meglio dai moderni PC!

Ho voluto aprire con questa considerazione per il semplice fatto che altrimenti rischiavo di essere richiamato alla realtà da parte dei lettori più attenti che magari il LISP lo usano ancora alla grande.

Bisogna considerare comunque che dal nostro punto di vista giudichiamo retrò un oggetto che non è più utilizzato nell'ambito home, proprio come appunto il linguaggio LISP.

Il LISP (LISt Processor) è un linguaggio funzionale con una struttura di base molto semplice (concetto che svilupperemo nel prosieguo di questo mini-corso), e quindi facile da implementare anche con risorse elaborative limitate, interpretato e dinamico. Tutte caratteristiche che lo rendevano adatto all'implementazione sulle piattaforme home anni '80.

Ce ne sono stati anche altri linguaggi che hanno conosciuto una sorprendente notorietà fra i possessori dei vari Apple, Commodore e Sinclair. Uno di essi è il Prolog, del quale si ricorda la famosa implementazione "Turbo" della Borland; un'altro è il Forth, astruso quel tanto che basta per scoraggiare i più, era comunque visto come un punto di arrivo per i "veri" programmatori, quasi al pari dell'Assembly.

Il vero successo del LISP è legato all'epoca d'oro dell'intelligenza artificiale. L'idea delle liste, come struttura dati, si presta molto bene ad una rappresentazione della conoscenza, per quanto grezza. Siccome ad un certo punto degli anni '80 sembrava che tutti dovessimo diventare programmatori di sistemi esperti e progettisti di "programmi intelligenti", era interessante disporre di una implementazione "light" del linguaggio da usare appunto sull'home di casa. Ormai tutto è superato da quello che è successo dopo il 1982, cioè l'ingegneria del software, la programmazione ad oggetti e così via.

Chi seguirà questa breve serie di articoli forse scoprirà (o riscoprirà) un linguaggio di programmazione un po' strano ma anche molto affascinante.

Del LISP ne esistono implementazioni per tutte le piattaforme, compresi Linux, Mac, Windows e tutti gli altri "esoterici" sistemi operativi che si trovano in giro. In passato si è visto il LISP sugli schermi dei PC home più diffusi. Chi vuole seguire con profitto le lezioni di questo corso dovrebbe dotarsi di un interprete LISP, qualunque esso sia, e seguire passo passo il codice e le spiegazioni.

Nota: ci sono differenze anche notevoli fra un "dialetto" e l'altro del LISP ma per gli scopi di base di questo corso gli esempi funzioneranno su qualsiasi implementazione.

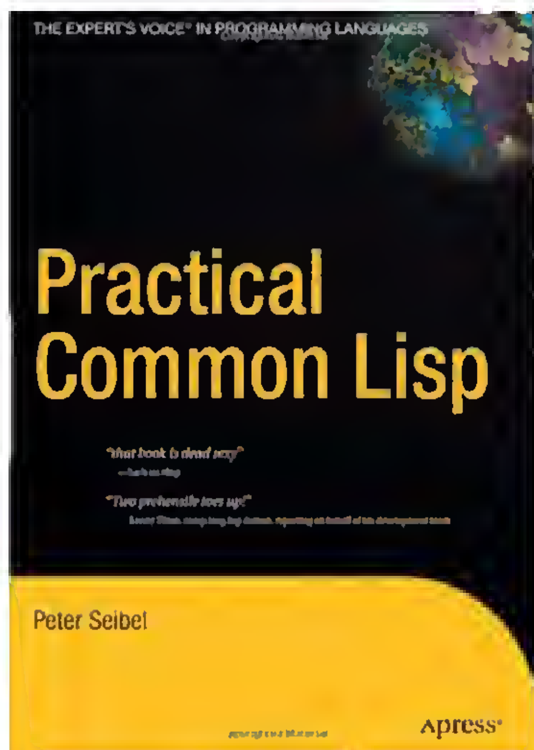


Una LISP Machine della Symbolics Inc. si presenta con una eleganza estetica straordinaria, oltre ad essere un sistema potente e facile da utilizzare.
Si tratta del modello 3610AE del 1986.

La versione del linguaggio più diffusa è la Common LISP, diventata il riferimento assoluto per il linguaggio grazie anche alla vasta letteratura che ne fa riferimento.

Per chi vuole esplorare seriamente il linguaggio, consiglio di partire dal link "http://www.fg-tech.it/index.php?option=com_content&view=article&id=57:lisp-come-iniziare&catid=34:Programmazione&Itemid=64" che spiega come reperire un compilatore free sotto Linux e come dotarsi degli altri tools indispensabili per lavorare agevolmente.

Chi invece vuole agire su un sistema retrò, è sufficiente che cer-



Practical Common Lisp, un testo che vale la pena consultare per imparare quasi tutto quello che serve per lavorare con il dialetto più diffuso di questo linguaggio.

chi un po' in giro per trovare un compilatore (più spesso un interprete) adatto alla sua piattaforma di elezione, sia esso un Apple II che un C64 che uno Spectrum.

Un'altro utile link è questo: "http://www.gigamonkeys.com/book/" dove si trova on-line il testo del volume "Practical Common Lisp", ottimo come introduzione al linguaggio. O ancora (per chi preferisce leggere in italiano) "http://it.wikibooks.org/wiki/Lisp", una pagina della Wikipedia curata dall'Italian Lisp Group che ha la sua base su: "http://www.lisp.it/".

Per ribadire il successo avuto dal linguaggio nei primi anni '80, riferiamo che furono costruiti dei computer dedicati, le cosidette "Lisp Machine". Questi computer, chiamati anche "Macchine simboliche", forse dal nome della ditta più famosa che ne costruiva una serie (la Symbolics), erano sistemi mono-tematici possiamo dire, cioè calcolatori progettati per essere degli ambienti di lavoro LISP e null'altro.

Le implementazioni più moderne del linguaggio sono dei compilatori nel senso che si sviluppa un sorgente come file di testo e lo si elabora attraverso un comando di compilazione. Nondimeno l'utilizzo dell'interprete è estremamente comodo per prendere confidenza con il linguaggio perché permette di provare "al volo" un esercizio o seguire una idea senza dover per forza passare per il ciclo classico: scrittura-compilazione-esecuzione.

Elementi base

Come ogni cosa che si deve imparare è bene partire dalle basi più elementari ed è quello che faremmo in questo paragrafo.

LISP è un linguaggio funzionale, il che in parole povere significa che non esistono veri e propri programmi, nel senso comune che si attribuisce a questo termine, cioè una sequenza di statement che vengono eseguiti a partire da un certo punto per arrivare ad una istruzione di stop. Un programma LISP in realtà è una raccolta di funzioni che vengono valutate ricorsivamente, nel senso che gli argomenti di una funzione possono essere altre funzioni e così via.

Programmare in LISP significa in pratica arricchire un certo ambiente di base con nuove funzioni fino ad ottenere la funzione "finale", cioè quella che effettua il calcolo o elaborazione che ci siamo proposti. Caricare o salvare un "programma" equivale a salvare un intero ambiente di lavoro come una sorta di dump binario della memoria o anche come testo contenente

tutte le definizioni di funzione approntate per l'elaborazione.

Alla fine si ha un risultato che può essere un numero o una stringa o più frequentemente una lista. La lista è la struttura dati sulla quale si basa tutto il linguaggio. Una lista è, lo si capisce dal nome, una sequenza di valori ordinati, come potrebbe essere in un'altro linguaggio un vettore unidimensionale. Per questa centralità della presenza delle liste e dal fatto che esse si rappresentano con le parentesi tonde, ad esempio (A B C), il LISP è definito anche "il linguaggio delle parentesi".

Gli elementi di una lista sono separati l'un l'altro da uno spazio (o da una tabulazione); non ci sono virtualmente limiti alla dimensione, anche se occorrerà tenere conto della memoria disponibile all'ambiente di esecuzione.

Un elemento in una lista può essere di tre tipi: è una informazione cosidetta "atomica", cioè un numero o una stringa, non ulteriormente divisibile; è a sua volta una lista che a sua volta può avere elementi che sono liste e così via in una ricorsione continua; è infine il risultato di una funzione che sarà un atomo o ancora una volta una lista.

Venendo alla definizione di atomo, diciamo che un Atomo Lisp è un elemento elementare del linguaggio che non può essere ulteriormente separato in componenti più elementari. In pratica sono atomi i numeri e le costanti numeriche, cioè le stringhe. I numeri (interi o reali) si rappresentano con le cifre, così come in qualsiasi altro linguaggio.

Ad esempio:

3.14 123 0.9 -12 +99.45 1.3E-7

Le costanti alfanumeriche sono racchiuse fra apici (doppi o singoli, dipende dall'implementazione) o anche da un solo apice che precede il primo carattere:

"MARIA" 'MARTA' 'MATTEO

In questo corso useremo la notazione con doppi apici sia perché è quella del Common Lisp e sia perché più comune ad altri idiomi di programmazione e anche perché in alcune implementazioni l'apice singolo è un operatore di costruzione delle liste.

Una lista si rapprensenta come una sequenza di elementi racchiusi da parentesi tonde:

(1 2 3)

("marta" "maria" "matteo" 12.5)

La separazione fra gli elementi di una lista è lo spazio o una tabulazione, più raramente viene sata la virgola o altri simboli come ":", etc...

Una lista può contenere altre liste:

("A" "B" (1 2 3) "pippo")

oppure funzioni (che sono poi a loro volta delle liste):

("zero" "uno" "due" (tre 3))

Qui abbiamo un elemento della li-

sta che è una funzione. In LISP le funzioni si rappresentano proprio così, come liste con in testa il nome della funzione e i parametri come altri elementi della lista stessa.

Come si vede una funzione si riconosce rispetto agli atomi perché il suo nome alfanumerico non è racchiuso da apici. Una stringa senza apici è una funzione o anche una variabile che rappresenta un valore.

Le variabili non sono un elemento indispensabile per lavorare con il LISP, ma sono state introdotte per comodità e per avvicinarsi al pensare comune di chi programma calcolatori. In effetti tutti i dati si possono conservare in liste dinamiche ma ovviamente assegnare un nome ad una lista è comodo e viene usato normalmente.

Primi approcci

Senza perderci in troppe chiacciere e sapendo che i nostri lettori vogliono venire subito al sodo, vediamo i primi esempi di utilizzo di un ambiente LISP.

Come si diceva LISP è un linguaggio funzionale, quindi qualsiasi "statement" (per parlare in termini classici) si realizza tramite l'implementazione di una funzione. Ogni funzione ha un nome, eventuali parametri e restituisce un risultato. Il risultato è un valore di tipo atomico o una lista oppure ancora un puntatore nullo (NIL in gergo Lisp-iano).

LISP è ricco di funzioni per la manipolazione di liste come ad esempio costruire una lista, estrarre il primo elemento, etc... Non mancano le operazioni aritmetiche, i confronti logici, i pattern iterativi e le funzioni trascendenti. Queste ultime usate poco, infatti chi si metterebbe a programmare in LISP un programma per il calcolo delle orbite dei satelliti di Giove?

Costruire una lista è questione di applicare una funzione agli argomenti in numero variabile:

```
(LIST 10 20 30 40 50)
```

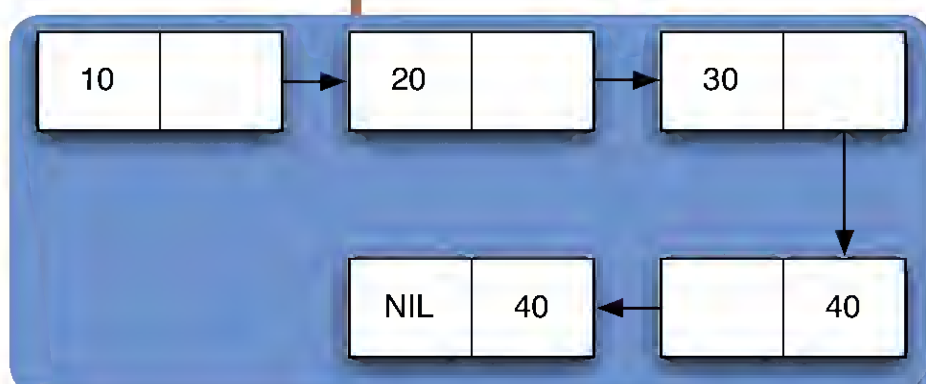
restituisce la lista (10 20 30 40 50).

In realtà cosa succede nel sistema? La funzione alloca la memoria necessaria a contenere una struttura dati di tipo lista, realizzata con coppie valore.puntatore, e istanzia le singole celle con i rispettivi valori. La funzione restituisce il puntatore al primo elemento della lista.

Si veda come esempio grafico la struttura schematizzata in figura 1.

Nel caso di costruzione di una lista con elementi alfanumerici al posto del valore la

Figura 1. Una lista in memoria



cella elementare contiene il puntatore alla stringa che è il valore dell'atomo elementare inserito nella posizione.

Esempio:

(LIST ""MARE" "LAGO" "MONTI")

ha una rappresentazione schematica come quella in figura 2.

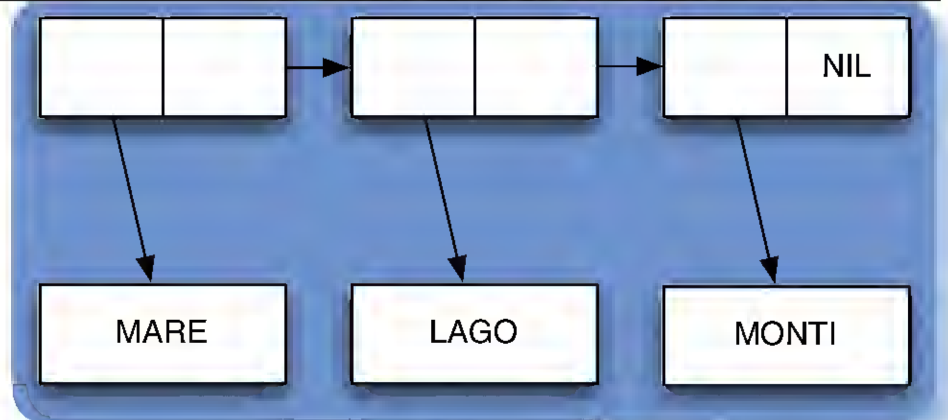


Figura 2.
Una lista con elementi alfanumerici

La seconda operazione basilare è quella di definire una variabile e il suo valore. Si fa con la funzione SET:

(SET 'PIPPO 12)

(SET 'PIGRECO 3.14)

Nota: l'apice singolo davanti ad un nome serve a bloccare il "quote" del nome. LISP infatti, essendo per sua natura un interprete ricorsivo, tende a valutare tutto quello che incontra. Se scrivessimo

(SET PIPPO 12)

l'interprete cercherebbe di valutate la parola PIPPO, cioè di sostituire al nome un suo valore che non esiste e che darebbe come risultatoi un errore di interpretazione.

Con le due definizioni sopra eseguite, abbiamo definito due variabili (in questo caso numeriche) che hanno nome PIPPO e PIGRECO. Tutte le volte che usiamo questi nomi in altre funzioni, LISP ne sostituirà il corrispondente valore.

Ad esempio:

(LIST PIPPO PIGRECO)

restituisce una lista di due elementi (12, 3.14).

Gli operatori aritmetici sono anch'essi inplementati come funzioni. Ad esempio per sommare due numeri:

(ADD 3 4) restituisce 7

(+ 3 4) è la stessa definizione in un dialetto diverso.

Nota: i moderni interpreti LISP tendono ad avere una sintassi c-like e quindi accettano gli operatori aritmetici nella forma simbolica consueta: + - * /; nel LISP originale le funzioni associate erano ADD, SUB, MULT, DIV; noi useremo prevalentemente questa seconda sintassi, anche se più prolissa per una questione di linearità del discorso.

Siamo giunti al termine di questa prima puntata del corso di LISP. La prossima volta cominceremo a lavorarci le liste: il LISP è nato per questo dopotutto.

[Sm]

TAMC

Algoritmi di SORT (parte 4)

Teoria e Applicazioni delle Macchine Calcolatrici: la matematica e l'informatica, le formule e gli algoritmi, la completezza e la computabilità, le strutture dati e tutto quello che sta alla base dell'informatica.

Riprendiamo il nostro viaggio lungo le strade che esplorano esaustivamente il problema di ordinare degli elementi.

La ricerca di una soluzione efficiente a questo basilare problema ha generato una serie abbstanza lunga di soluzioni che hanno coinciso con il relativo algoritmo implementativo. Ricordiamo il problema fondamentale: dato un vettore di interi ottenere al termine dell'elaborazione un vettore contenente gli stessi elementi di partenza ma ordinati in ordine crescente.

Il Merge Sort

Analogo al QuickSort deriva da esso ma fa propria la seguente osservazione fondamentale: se siamo in possesso di due vettori ordinati, costruire un vettore "fusione" dei due è semplice: si tratta di scorrere i due vettori e inserire nel vettore risultante l'elemento minore. Spingendo al limite il ragionamento è banale affermare che un vettore di un solo elemento è ordinato per definizione, quindi perché non dividere il vettore originale in N parti, ognuna di un solo elemento e poi procedere al merge?

Questo in parole povere la strategia che sta alla base del Merge Sort. Notiamo fra l'altro che questo algoritmo ha un utilizzo molto frequente nei casi di ordinamento di insiemi particolarmente grandi che richiedono una strategia di divisione del problema in pezzetti più piccoli, magari appoggiati sul file-system, e poi "fusi" per restituire il vettore finale ordinato.

Mergesort somiglia molto al Quicksort che abbiamo discusso nella puntata precedente di questa serie (si rimanda per approfondimenti al fascicolo n. 20 del dicembre 2008... cavolo come passa in fretta il tempo!).

Il vettore originale viene diviso prima a metà e iterativamente in sottoinsiemi sempre più piccoli fino a giungere a N insiemi di un solo elemento. Fino a questo punto l'algoritmo fa quello che faceva il Quicksort, con l'unica differenza che mentre nel Quicksort c'era anche uno spostamento di elementi da una parte all'altra del sotto-vettore in relazione alla loro cardinalità rispetto all'elemento pivot, nel Mergesort non si assiste a nessun spostamento e il vettore rimane inalterato.

Alla fine della divisione, realizzata attraverso una funzione ricorsivamente chiamata sui due sottoinsiemi cui viene diviso il sotto-vettore, c'è la fase di merge attraverso un backtrack iterativo che raccoglie i sotto-vettori già ordinati e restituisce un unico vettore risultante.

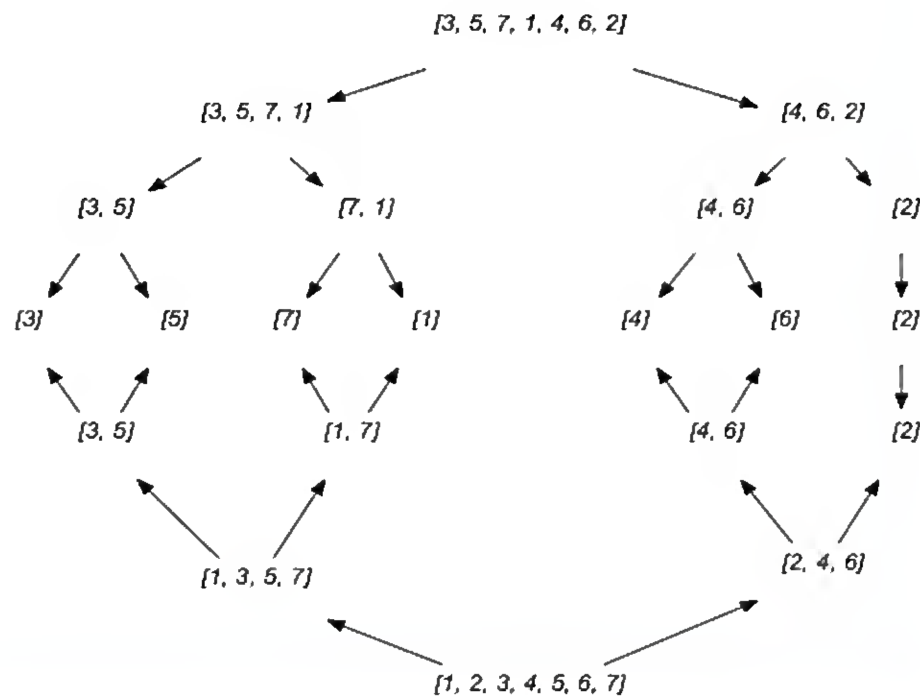


Figura 1. L'albero risultante dalla divisione del vettore in sottovettori e la ricostruzione per parti del vettore ordinato.

Seguiamo l'algoritmo sul vettore di esempio che usiamo nel nostro studio:

[3, 5, 7, 1, 4, 6, 2]

In figura 1 lo schema seguito dalla funzione Mergesort sul vettore di partenza.

In realtà l'algoritmo non aspetta di avere tutto l'albero per procedere al merge, ma processa le coppie di sotto-vettori a mano a mano che esse diventano disponibili.

Nel listato mostrato nella pagina seguente abbiamo riportato l'intero programma scritto in Pascal (Borland Pascal 7.0 per la precisione).

Come si noterà seguendo il listato abbiamo due funzioni: la prima "merge" si occupa di fondere i sottovettori in un'unica sequenza ordinata che sarà sostituita nel vettore originale allo stesso posto occupato dai vettori di partenza; la seconda "mergesort" è la procedura ricorsiva che chiama se stessa sulle due metà del vettore che sta processando e chiama "merge" non appena i due sotto-vettori sono disponibili alla fusione.

Si rimanda alla figura 2 per l'output risultante dal programma.

La funzione "mergesort" è banale, visto che semplicemente deve decidere dove mettere l'elemento di separazione del vettore. Da questo punto di vista Mergesort risparmia tempo rispetto al Quicksort, soprattutto nel caso peggiore (vettore di partenza ordinato a rovescio), vi-

```
Program sortmerge;

var
A, B : array[1..7] of integer;

procedure merge(p, q, r: integer);
var
i, j, k, s : integer;

begin
    i := p; j := q + 1; k := i;
    while (i <= q) and (j <= r) do begin
      if ( A[i] < A[j]) then begin
        B[k] := A[i];
        i := i + 1;
        end
      else begin
        B[k] := A[j];
        j := j + 1;
        end;
      k := k + 1;
      end;

    if (i <= q) then begin
      for s := i to q  do begin
        B[k] := A[s];
        k := k + 1;
        end
      end
    else begin
      for s := j to r  do begin
        B[k] := A[s];
        k := k + 1;
        end;
      end;
    for i := p to r do begin
      A[i] := B[i];
      end;
    writeln(A[1], ‘ ‘, A[2], ‘ ‘, A[3], ‘ ‘, A[4], ‘ ‘, A[5], ‘ ‘, A[6], ‘ ‘, A[7]);
end;

procedure mergesort(p, r : integer);
var q : integer;

begin
    if (p < r) then begin
      q := (p + r) div 2;
      mergesort(p, q);
      mergesort(q + 1, r);
      merge(p, q, r);
      end;
end;

begin
  A[1] := 3;
  A[2] := 5;
  A[3] := 7;
  A[4] := 1;
  A[5] := 4;
  A[6] := 6;
  A[7] := 2;

  writeln(‘Merge Sort’);
  writeln;
  writeln;
  writeln(A[1], ‘ ‘, A[2], ‘ ‘, A[3], ‘ ‘, A[4], ‘ ‘, A[5], ‘ ‘, A[6], ‘ ‘, A[7]);
  writeln (‘===============’);
  mergesort(1, 7);

  writeln (‘===============’);
  writeln(A[1], ‘ ‘, A[2], ‘ ‘, A[3], ‘ ‘, A[4], ‘ ‘, A[5], ‘ ‘, A[6], ‘ ‘, A[7]);
  readln;

end.
```

sto che Quicksort deve decidere sull'elemento pivot che separa i due sotto-vettori.

Più complessa la funzione "merge" che può essere divisa in tre corpi principali:

Il primo while realizza l'algoritmo chiamato "di inseguimento", un pattern tipico quando si debbano fondere due insiemi.

L'algoritmo si chiama così perché i due puntatori i e j agli elementi rispettivamente del primo e del secondo vettore, si "inseguono" incrementandosi ora l'uno ora l'altro in base ai valori degli elementi contenuti nei vettori stessi.

Il secondo corpo della funzione "merge" è il residuo dell'inseguimento. Dal momento che l'algoritmo porta in fondo uno dei due indici, ne segue che tutti gli elementi residui dell'insieme non esaurito vanno semplicemente copiati nel vettore fuso.

Infine il terzo settore della funzione copia il sotto-vettore nella posizione corrispondente del vettore originale (copia di B in A).

Questo è necessario perché un sotto-vettore ordinato uscito dall'applicazione della funzione "merge", sarà preso in considerazione dalla stessa funzione nella fusione di due sotto-vettori allo stesso livello di backtrack.

Sembrerebbe che la presenza del vettore di appoggio B e le

continue copie di elementi da un vettore all'altro, possano rallentare l'algoritmo. Ovviamente è così e aggiungiamo subito che Mergesort soffre di una necessità di spazio doppia rispetto al Quicksort. Le copie dei sotto-vettori sono però ottimizzate (si muovono solo il numero minimo di elementi.

Analizzando l'ordine algoritmico si calcola facilmente che il Mergesort ha la stessa complessità del Quicksort: $n\log_2(n)$ ma è la stessa anche nel caso peggiore, mentre Quicksort nei casi sfortunati arriva ad un n^2 decisamente "pericoloso" per insiemi con elevati elementi.

Infine ricordiamo quello che abbiamo detto all'inizio: Mergesort si presta bene all'ordinamento di record su file, anche se in questo caso non si spinge l'algoritmo fino ad insiemi di un solo elemento, che genererebbero una proliferazione di file di appoggio, ma si mescola Mergert con qualcosa di efficienza comparabile su piccoli insiemi di elementi.

Bene, ci siamo impadroniti anche di questa tecnica di ordinamento. La prossima puntata ne troveremo un'altra, parimenti interessante: l'Heapsort.

Alla prossima

[Sm]

```
Merge Sort

3 5 7 1 4 6 2
================
3 5 7 1 4 6 2
3 5 1 7 4 6 2
1 3 5 7 4 6 2
1 3 5 7 4 6 2
1 3 5 7 2 4 6
1 2 3 4 5 6 7
================
1 2 3 4 5 6 7
```

Figura 2. L'output del programma sul vettore di esempio..

```
    if (i <= q) then begin
       for s := i to q  do begin
          B[k] := A[s];
          k := k + 1;
          end
       end
    else begin
       for s := j to r  do begin
          B[k] := A[s];
          k := k + 1;
          end;
       end;
    for i := p to r do begin
       A[i] := B[i];
       end;
end;

procedure mergesort(p, r : integer);
var q : integer;
```

Figura 3. Il programma in esecuzione sotto debug nell'ambiente BP 7.0

L'intervista

Vari personaggi e amici incontrati qua e la per una chiacchierata sul mondo del retro computing.

Intervista ad Albert



Albert (non è il suo vero nome, ma ci ha chiesto di essere nominato con questo nick) è una persona che lavora presso una università nel nord Italia come informatico dal 1987. Appassionato da sempre di calcolatori si è dedicato da una decina d'anni a questa parte al recupero di sistemi di elaborazione, per la maggior parte della linea Sun Microsystem. Ha accettato di scambiare con noi qualche battuta via posta elettronica sulle sue esperienze e sulle sue idee riguardo il retro computing.

Redazione.

Albert, sappiamo che sei all'università da più di vent'anni, di cosa ti occupi esattamente?

Albert.

Sono un tecnico di dipartimento e lavoro nel laboratorio del dipartimento di informatica. Abbiamo una infrastruttura piuttosto composita: server di vario genere, servizi Internet, macchine per il calcolo parallelo, etc... Praticamente passo la mia giornata in sala macchine o nel laboratorio ad aiutare gli studenti che fanno la tesi o il dottorato: preparo le macchine, installo il software, gestisco gli account e curo il sito web del dipartimento e dei gruppi di ricerca.

Redazione.

Un lavoro molto bello per un informatico, credo tu ne convenga.

Albert.

Sì, naturalmente. Si tratta di un ambiente molto stimolante perché ci sono sempre cose nuove da imparare e qui non si lesina certo sui mezzi tecnici, per cui in vent'anni ho messo le mani sui sistemi più moderni e costosi che sono via via usciti sul mercato. Poi c'è l'aspetto ricerca alla quale partecipo, seppure marginalmente e si conoscono molti personaggi famosi, dei veri cervelloni. Ad esempio ricordo una visita di Minsky che si fermò da noi per un mese e più e fece un ciclo di lezioni e di seminari molto seguiti e interessanti.

Redazione.

Hai lavorato anche nel privato o sei approdato subito in università?

Albert.

Ho lavorato prima all'IBM, subito dopo la laurea e poi alla Olivetti a Milano. In IBM posso dire che non facevo nulla! E' strana (o almeno lo era) questa azienda: prendeva i giovani neo-laureati e li metteva a... non fare niente. Sì qualche corso, ma poco altro. Forse mi sono stufato prima che decidesssero dove mettermi a lavorare ma ci sono stato quasi un anno del tutto inoperoso!

In Olivetti sono approdato dopo un paio d'anni alle dipendenze di un negozio che seguiva le installazioni dei registratori di cassa e cose di questo genere, diciamo "da ufficio". In Olivetti seguivo le installazioni e le riparazioni dei mini presso i clienti. Ci sono stato bene per quasi sette anni, poi ho deciso di cambiare aria e tornare da dove ero venuto (sono laureato in Scienza dell'Informazione, come si chiamava allora).

Redazione.

Parlaci di retro computing. Come lo vedi, qual'è il tuo punto di vista, etc...

Albert.

Beh, a mano a mano che i sistemi diventavano obsoleti il dipartimento ne comprava di nuovi, così le macchine dismesse finivano in magazzino e in vari sottoscala e bugigattoli sparsi in giro nel palazzo. Con i beni statali non è facile procedere alla loro alienazione, ad esempio non si possono vendere e regalarli comporta trafile burocratiche immense, per cui generalmente si preferisce accantonarli e basta.

Così verso il 2000 ci trovammo pieni di una accozzaglia di sistemi incredibile. Il fatto è che fino al 1990 le macchine avevano una vita relativamente lunga, mentre dopo scoppiò la stagione dei PC a basso prezzo che però si cambiavano anche ogni due anni!

Quando il dipartimento decise di realizzare una mostra sulla storia

dell'informatica in vista del 2000 appunto, aprimmo questi "magazzini" e cominciammo a ripassare le vecchie macchine per scegliere quelle da esporre.

Di questo progetto mi occupai principalmente io, assieme ad un collega che lavorava ad Ingegneria. Passammo sei mesi a rimettere in sesto decine di calcolatori. Alla fine la mostra ebbe un successo notevole, anche grazie ai contributi che fecero altri per quanto riguarda i sistemi Apple e gli home computer, che in dipartimento ovviamente non sono mai entrati :-)

Forse non è proprio del tutto vero in effetti... Ricordo un progetto di rivelazione di segnali fatto con un Atari ST (credo un 1040 ma non ne sono sicuro), che poi il ricercatore si portò via non so dove, credo che lavorasse in un ospedale e i segnali che elaborava erano appunto dei segnali provenienti da sensori attaccati ai pazienti.



Redazione.

Quindi niente home, ma qual'era la specialità, se possiamo chiamarla così?

Albert.

In dipartimento si lavorava quasi esclusivamente con macchine della Sun Microsystem. C'era qualche MAC della Apple e cloni PC ovviamente. Però si lavorava quasi esclusivamente su Unix come sistema operativo, cioè Solaris sulle Sun, Xenix o Linux sugli altri e pure sui MAC con AIX. Poi c'erano sistemi più esoterici come microVax della Digital, Symbolics, etc...

Credo di aver visto tutti i modelli di Sun Workstation prodotti e di aver installato tutte le versioni di Unix prima e Solaris poi che la Sun ha sviluppato.

Redazione.

Ecco il perché della tua propensione per i sistemi della Sun.

Albert.

Infatti. Dopo il progetto dell'esposizione per il 2000, continuai a mantenere efficienti queste macchine che avevano trovato rifugio in una saletta del dipartimento. Ce n'erano però molte altre che si sarebbero potute recuperare e che giacevano nei sottoscala. L'occasione si presentò in seguito ad un trasloco che il Dipartimento dovette fare nel 2002. Non era proprio il caso di portarsi appresso i vecchi "catorci" (lo dico

con affetto), così ci si decise per l'alienazione. In questa occasione mi portai via una trentina di sistemi fra Sun, Mac e altro riempiendo fino al soffitto una vecchia rimessa che avevo in campagna a pochi chilometri da Ivrea. Non avevo una idea precisa di cosa ne avrei fatto, ma mi sembrava un peccato mortale che finissero in discarica. Pensavo anche di rivenderle ma poi in realtà finì per regalarne molte e recuperare da altre la pura spesa del trasporto.

Tre anni fa circa ristrutturai la casa dei miei genitori in campagna e così potei disporre di uno spazio decente da trasformare in sala hobby/laboratorio. Che poi è il mio "museo privato", diciamo così...

Redazione.

Ma poi hai raccolto anche altri pezzi da collezionare o ti sei limitato a questo primo "lascito"?

Albert.

In effetti c'è stato un periodo durante il quale ho frequentato i vari mercartini, fiere dell'elettronica e eBay per scambiare/recuperare qualche vecchio home o poco più. E' difficile trovare un privato che abbia un Digital Raimbow da vendere... più spesso trovi i Commodore 64 o gli Amiga, come tu sai benissimo.

Redazione.

E i PC?

Albert.



Ne sono strapieno, come tutti immagino. Non tengo molto a cuore questi sistemi, così giusto qualcuno per evidenziare una evoluzione dell'informatica e del costume (perché no?).

Il mio obiettivo/sogno è fare un vero museo, magari piccolo, ma che sia il più possibile completo riguardante l'evoluzione dell'informatica. Penso che non ci riuscirò mai, ma intanto è un'idea che mi permette di continuare a lavorare a tempo perso per recuperare le vecchie macchine.

Redazione.

Altri progetti, idee?

Albert.

Sì, attualmente sono in contatto con un museo privato in Francia che sta raccogliendo volontari e professionisti in grado di curare un settore di una collezione dedicata ai sistemi di calcolo. Dovrei occuparmi dei sistemi mini e delle workstation Unix. Se la cosa va in porto credo sarà una buona occasione per mettere a frutto le mie conoscenze in questo ambito e fornire un servizio in favore della conservazione della memoria.

Redazione.

Molto bello, ma ti pagano?

Albert.

Per ora ci sono andato due volte e mi hanno pagato le spese. Il museo si trova a Lione, non proprio dietro l'angolo e non posso certamente permettermi di rimetterci anche le spese!

Redazione.

Questa cosa è interessante. Da tempo sosteniamo anche noi una via professionale come sbocco all'hobby del retro computing, cosa peraltro normale in altri settori. Cioè l'idea che si possa lavorare a tempo pieno o anche parzialmente per una istituzione di conservazione e venire anche pagati.

Albert.

Sono d'accordo. Nonostante ci siano in giro moltissime persone che potrebbero fornire un aiuto professionale nel campo, spesso sono tenute lontano proprio dalla "miseria" di certe proposte di collaborazione. Penso che siamo ancora all'inizio in questo campo ma che ci si dovrebbe decidere nel giro di una ventina d'anni, altrimenti si rischia che le persone non ci siano più per davvero!

Redazione.

Già, proprio come dici tu. Molti hanno l'idea di trasformare la loro collezione in museo, ma poi la abbandonano. Secondo te quali sono le difficoltà? Tu ci avrai sicuramente fatto dei ragionamenti.

Albert.

In primis lo spazio. Disporre di uno spazio adeguato e soprattutto adatto ad accogliere una esposizione permanente è la difficoltà maggiore. Non vedo altra strada che sia il comune o un ente pubblico a mettere a disposizione questi locali. Direi che basterebbero anche cento metri quadrati di spazio ma in un ambiente luminoso, pulito e che offra una dotazione minima di servizi: corrente, riscaldamento per l'inverno e bagno. Solo così si potrebbe partire con attività come visita di scuole, dimostrazioni, serate informative, etc...

Io non ho abbandonato l'idea e anzi credo che questa esperienza francese, se andrà in porto, mi sarà utilissima per progettare qualcosa di valido. Penso che mi ci dedicherò fra una decina d'anni quando forse andrò in pensione.

La mia idea di spazio espositivo è però atipica rispetto al classico museo. Secondo me quello che sarebbe utile realizzzare è un ambiente informatico a tutto tondo dove si svolgano corsi, manifestazioni etc... e il retro computing sia solo una parte di questo progetto.

Redazione.

Bhe, una idea affascinante ma certo difficile da realizzare, soprattutto da soli...

Albert.

Infatti sto tenendo i contatti con vari amici che si sono dimostrati interessati. L'unico dubbio è che la burocrazia sia talmente cavillosa che alla fine ci costringa nostro malgrado a rinunciare.

Redazione.

Ma l'università, non potrebbe essere l'ente che citi come partner ideale per la realizzazione della tua idea?

Albert.

Sì e no. Magari riuscirei a convincere un direttore di dipartimento o il preside della facoltà a realizzare questo spazio espositivo, ma so come vanno le cose nell'accademia: non appena viene nominato un nuovo preside, questi mette in discussione subito tutte le iniziative di chi l'ha preceduto... Eh, ne ho visti tanti di contro-ordini in venti anni che sono in università!

Redazione.

Chiudiamo qui questa conversazione ma devi prometterci che ci farai sapere subito se parte qualche tua iniziativa di museo o esposizione. Promesso?

Albert.

Promesso, senz'altro. E vi manderò anche quei pezzi per farne degli articoli sul mondo Sun, ma non troppo presto... ho un sacco di cose da fare in questo periodo.

Ciao a tutti e buon retrò, qualsiasi cosa significhi per voi.



Jurassic News

La prima e unica rivista di retro computing disponibile in Italia dal 2006









